Sensor Distribution for Collaborative Localization using Radio Ranging

by

Benjamin Alfred S. Werner

Submitted to the Department of Aeronautics and Astronautics
in partial fulfillment of the requirements for the degree of

Master of Science in Aeronautics and Astronautics

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

[September]
August 2008

Author .....
Department of Aeronautics and Astronautics
August 21, 2008

Certified by .....
Nicholas Roy
Assistant Professor of Aeronautics and Astronautics
Thesis Supervisor

Certified by .....
Megan Mitchell
Member of the Technical Staff, C.S. Draper Laboratory
Thesis Supervisor

Accepted by .....
David Darmofal
Chair, Committee on Graduate Students

Sensor Distribution for Collaborative Localization using Radio Ranging

by

Benjamin Alfred S. Werner

Submitted to the Department of Aeronautics and Astronautics
on August 21, 2008, in partial fulfillment of the
requirements for the degree of
Master of Science in Aeronautics and Astronautics

Abstract

This thesis explores the localization of a group of networked agents using range measurements between themselves in a global reference frame. While operating in an environment with sparse Global Positioning System availability and intermittent inter-agent range measurements, additional sensors may be needed to maintain a given level of position accuracy. This research explores the balance between penalties associated with the addition of sensors and the ability to localize all agents to a specified accuracy. The problem is defined as an optimization formulation that minimizes the cost of additional sensors over the group while requiring accurate positioning knowledge for all agents. The first result of this thesis is a novel method for solving the posed optimization problem. This method avoids searching all possible instrumentations by exploiting structure in the problem: testing a single sensor configuration for localization accuracy sometimes allows for implicit elimination of multiple configurations. Discerning the best configuration to test for localization accuracy decreases the required search time to solve the optimization problem. The second contribution of this thesis comes from the application of the optimization's search procedure to problem of distributing inertial measurement units to a group of agents. The effects of various environmental conditions on the required distribution of inertial measurement units are investigated.

Thesis Supervisor: Nicholas Roy
Title: Assistant Professor of Aeronautics and Astronautics

Thesis Supervisor: Megan Mitchell
Title: Member of the Technical Staff, C.S. Draper Laboratory

Acknowledgments

Thanks to...

Megan, whose extensive experience was an irreplaceable element of this thesis work. As a mentor in both engineering and professionalism, her positive outlook toward inspiring possibilities was always uplifting, and her integrity is admirable. I also thank Professor Roy, with his incredible insight into any problem presented to him. His prompting to turn my project towards a new and unsolved problem stretched my abilities and produced results that I am happy to have achieved. Long after having left MIT, I will still be learning from what my two advisors taught me. Draper technical staff, especially Matt Bottkol, for their availability to answer any conceivable question about navigation, Dane Richter and Peter Shermin, who always encouraged my interest, and Brent Appleby for his guidance in crucial moments.

Peak, who, besides making it possible for my thesis to be turned in, was a great cubicle-mate. Paul, with whom I enjoyed a good 80% of my day with for the last nine months at MIT. Josh, with whom my conversations, no matter how simple, will always end with a long discussion of philosophy that must be continued later. Sam, with whom I have enjoyed many a discussion of filtering, and whose ability to warm any environment will be missed.

Doreen, for her cheerful laugh and sincere encouragement that carried me through.

My grandparents, for being supportive in the best ways: good advice, a great home, the more than occasional puns, and oatmeal chocolate chip cookies. My sister Rachel, who as a candidate for a PhD degree in psychology, is the only person in the family that is in an authoritative position to convince me that my various neuroses are completely normal. My sister Martha, for putting up with a pretty nerdy older brother. My brother Daniel, for making sure I made it to this point without fatally injuring myself during childhood.

Finally, I wish to dedicate my work to my mother and the memory of my father, both of whom have always put my growth and education before any personal goal, giving me the best even as difficult as I made it sometimes.

August 21, 2008

This thesis was prepared at the Charles Stark Draper Laboratory, Inc., under the Independent Research and Development Project Numbers 22026-001 and 21137-001, titled "Dismount Geolocation System". Draper Laboratory Report Number T-1612.

In consideration for the research opportunity and permission to prepare my thesis by and at The Charles Stark Draper Laboratory, Inc., I hereby assign my copyright of the thesis to The Charles Stark Draper Laboratory, Inc., Cambridge, Massachusetts.

Publication of this thesis does not constitute approval by Draper Laboratory of the findings or conclusions therein. It is published for the exchange and stimulation of ideas.

Contents

THIS PAGE INTENTIONALLY LEFT BLANK

List of Figures

List of Tables

THIS PAGE INTENTIONALLY LEFT BLANK

List of Algorithms

THIS PAGE INTENTIONALLY LEFT BLANK

Chapter 1

Introduction

1.1 Background and Motivation

In 1999, firemen were dispatched to respond to a fire at a cold storage facility in Worcester, Massachusetts. When informed that the building may have still been occupied, several groups attempted a search and rescue operation in the building. While inside, six of the firemen became disoriented due to near zero visibility, eventually running out of breathable air. In emergency response situations such as this, it is often the case that a group of people collaborate to achieve a shared goal. Whether it is the case of firemen finding their way through a burning building, or military personnel clearing an urban area, coordination is imperative in safely achieving the goal. In such situations, knowing the location of one's team members is essential for monitoring of progress and reacting to situations that present themselves.

Military leaders recognize the importance of information that helps formulate a common operational picture, citing it as one of the elements of organized power. Further, synchronization, defined as the arrangement of activities in time, space, and resources, is listed as a fundamental tenet of operations [34]. Thus, as technology becomes available, advancing localization algorithms for both people and objects in emergency and combat situations has proven to be an important research topic.

To address situations similar to the scenario at the cold storage facility and others where localization of human agents is imperative to success, research institutions and

commercial companies have worked to apply techniques for localization using range-based measurements from Ultra-wideband (UWB) radios for indoor localization and tracking [25]. Many range-based techniques use the propagation of a signal, whether it is ultra-sonic, radio, or light, to measure distance. Some methods additionally use a returning signal to measure the distance. Most work has been focused on defining two distinct groups of agents: stationary beacons and dynamic agents that must be tracked [28]. However, it will be shown that ranging measurements can also be used without such constraints in collaborative localization with peers.

Collaborative localization has been defined as the act of determining the position of agents in their environment and then tracking them as their position changes in time, taking measurements from multiple agents and sharing them. Through the sharing sensor data amongst the agents, localization performance can be enhanced [13]. The entire group can benefit from one agent having a high quality sensor; furthermore, complementary sensor information is shared between users.

1.2 Statement of Objective

This thesis explores the problem of geolocation for a group of communicating agents with a variety of sensors available to them. Given this network of agents capable of inter-user range measurement and the sharing of additional sensor information, it is desired to find the minimal sensor configuration required to successfully localize every agent in a global coordinate frame. To this end, the work presented in this thesis is represented in three major steps: (1) the formulation of an optimization problem that captures the requirements for the instrumentation of a sensor network for localization and appropriately prefers lighter and less costly instrumentations, (2) the introduction of a feasible and efficient algorithm for solving that optimization problem, and (3) the verification of the optimal solution through the use of stochastic simulation. Using the verified solutions from the optimization problem, the effect of changing a set of mission and optimization parameters for the problem of instrumenting a group of networked agents with inertial measurement units is investigated.

In the following subsections, a description for each of these steps is introduced, providing motivation and reasoning for the method of pursuit. An enumeration of the main contributions of this work can be found in Section 1.3. The chapter is concluded with Section 1.4 outlining the organization of the rest of the thesis.

1.2.1 Sensor Configuration Optimization

In open areas, the localization and tracking problem has largely been addressed by the tracking of ranging signals from the Global Positioning System (GPS) satellites. The GPS constellation guarantees access to sufficient information to localize in such an environment by allowing its users to know the exact positions of the satellites, as well as the range to those satellites. This information is usually enough to satisfy the requirements of emergency and combat personnel. However, since GPS signals are weak when they reach the surface of the earth, they are easily occluded by tall objects in the immediate area. For this reason GPS signal coverage is limited in cities. Indoors, infrequent and degraded GPS coverage can be expected. In these difficult environments, localization is increasingly important for many operations.

In collaborative localization, inter-agent ranging is able to help when a subset of the agents have externally acquired information about to their global location. Ultra-wideband (UWB) technologies are a promising technology for radio ranging as well as the broadcasting of other localization information. UWB signals have the advantage of penetrating materials that GPS cannot near the surface of the earth. This is a result of the larger bandwidth transmitted and higher power levels at the receiving end for UWB radio. Although these benefits make UWB attractive for localization, the inter-agent range that UWB measurements provide cannot solely be used for localization in a global frame. Inter-agent ranging measurements do not provide any information about the rotation and translation of the entire network of agents in a global frame unless additional measurements such as GPS are available. The optimization presented in this thesis optimization considers the availability of both UWB and GPS measurements, and their strengths and weaknesses.

A large subset of the problems associated with localization and subsequent track-

ing of agents in a global frame can be mitigated by using accurate and expensive sensors. Given unlimited resources and ignoring operational constraints such as weight, the answer for instrumenting a network of agents for localization becomes obvious: all agents should be given a full suite of the best sensors available in order to have the greatest chance of successfully tracking their position. As an example, given a properly calibrated strategic grade inertial measurement unit (IMU), an agent can travel for an hour without any other sensing devices and acquire a position error on the order of 10 meters [28]. Given the weight and cost of such a unit, deploying this IMU is prohibitive in most scenarios. The necessity of choosing the appropriate sensors is apparent.

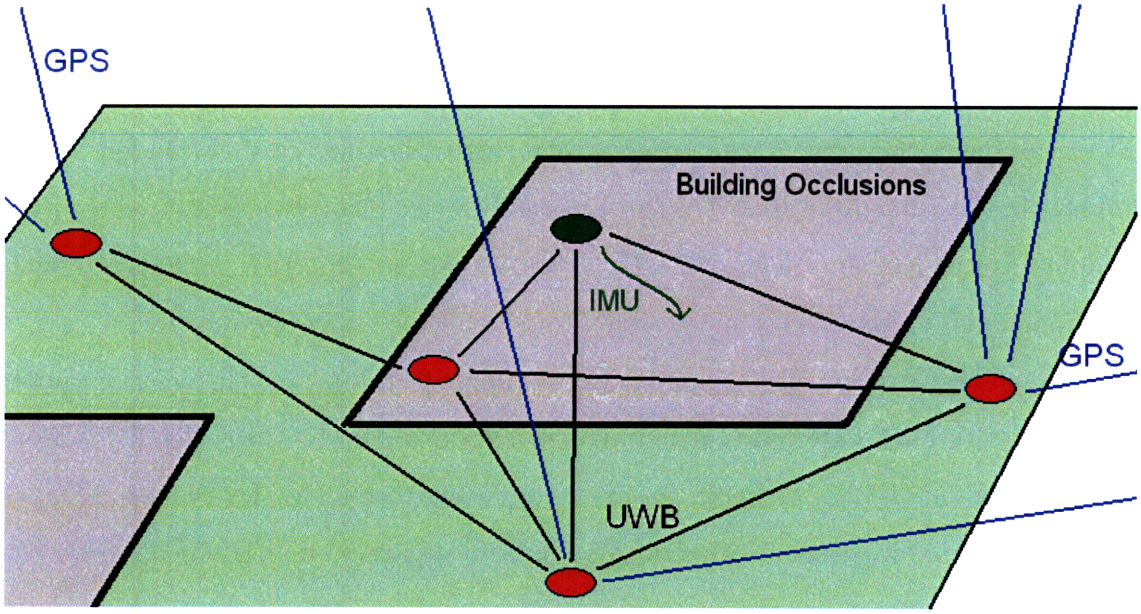

Figure 1-1: Multiple sensors among users aid localization by sharing complementary information.

In formulating the problem, a cost function is required to indicate the preference in the configuration of aiding sensors. For the purpose of this study, a few different grades of IMUs were considered for localization aiding, although the algorithm introduced is capable of handling a mix of additional sensors. The specific cost function used in this thesis is a sum of a representative price of each IMU that was added to the group.

For each optimization problem, the trajectories of the agents and the expected

sensor measurements along the trajectories are simulated, and the posterior position accuracies are calculated for various configurations of sensors. For these configurations, the position accuracies at every step in time are tested against a required accuracy. Computing the optimal configuration of sensors is a matter of searching through the possible instrumentations of the network to find the lowest cost that satisfies the accuracy requirements. Accuracy requirements take the form of restrictions on diagonal components of the position error's covariance matrix. This matrix is calculated at every timestep using the Kalman filter equations for propagating covariance forward.

1.2.2 Search Algorithm Description

As the number of network agents and instrumentation choices grow, the number of configurations to evaluate grows exponentially. Iteration through all possible instrumentations of the sensor network would therefore be prohibitively computationally intensive. Further, the covariance propagation test for a single instrumentation is also computationally expensive.

Two observations about the structure of the optimization problem allow an absolute minimum cost configuration to be reached without executing a covariance propagation for every configuration: (1) performance relations can be established between configurations that allow a single failed configuration test to eliminate a large group of configurations that are also guaranteed to fail to meet the required accuracy constraints, and (2) cost relations can also be established that allow a single passed configuration test to eliminate a large group of more costly configurations from being the candidates for minimum configuration.

Given that a number of configurations will be eliminated on a single test, the problem then becomes choosing configurations in order to minimize the number of required covariance propagation tests. This thesis focus is on choosing the next configuration that will eliminate the most configurations at every step. Even though this search algorithm does not test every possible configuration, it is shown that the configuration yielded will still be optimal.

1.2.3 Stochastic Simulation and Kalman Filter

It is important to test whether the minimal cost configuration as specified by the original optimization problem will actually meet the criteria on the covariance when subjected to stochastic measurements. Thus, a stochastic simulation is created for each sensor, based on the original probabilistic model used in the covariance test.

The purpose of this simulation is to reveal any deviations in a Kalman filter's covariance from the optimization's predicted covariance, where information was derived from truth data as opposed to being subject to errors in linearization that occur in a Kalman filter accepting stochastic measurements. Additionally, it provides the verification that the error in the position estimate falls within the probabilistic model assumed by the optimization's predictions.

1.3 Contributions

The following list is that of contributions from this research project:

1. the formulation of an optimization problem in order the minimize the cost of a set of aiding sensors required to successfully localize every agent in a network through a given environment and trajectory, a novel search algorithm for solving the optimization, and an analysis of the runtime of the different search strategies,

2. the verification of the results from the covariance tests executed by the above optimization with a Kalman filter integrating simulated stochastic measurements for each sensor used by the network, and

3. the presentation of the effects on the optimal network instrumentation yielded while varying environmental factors as well as optimization parameters.

1.4 Chapter Organization

The chapters of this thesis are organized in the following manner. Chapter 2 provides an overview of related work. The basic concepts of probabilistic inference and algorithms that are used in the optimization procedures as well as the stochastic simulation are reviewed. The principles of operation for all sensors used in the optimization are discussed. Finally, current work in the fields of range-based and collaborative localization upon which this thesis is based is also reviewed.

In Chapter 3, the sensor placement optimization cost function, constraints, and search space are defined. Observations about the problem's structure, along with the approach to solving the optimization problem are then discussed. Supporting procedures such as the method for searching the problem space, and a covariance propagation procedure are detailed. Results from several optimization scenarios are shown and discussed. During the discussion of the covariance propagation, measurement and propagation models are defined. Chapter 3 also discusses the effect of choosing different methods for searching the problem space on the algorithm run-time and space complexity.

Chapter 4 will discuss the rationale behind the generation of maps and trajectories for simulating indoor, urban, and open area environments. Further, the testing strategies for these environments to determine a viable instrumentations will be explained. Results from the testing strategies are then displayed and analyzed to show the requirements on sensor configurations. For the verification of the optimization's resulting configurations, a stochastic measurement generator and Kalman filter implementation is shown. A statistical analysis on the mean and covariance is used to show that the result under stochastic processes is almost identical to that predicted by the optimization.

In conclusion, Chapter 5 enumerates future work, summarizes the results from previous chapters, and reviews the contributions made.

THIS PAGE INTENTIONALLY LEFT BLANK

Chapter 2

Probabilistic Estimation and Localization

2.1 Chapter Overview

The process of localization and tracking involves using measurements taken from sensors in one's environment to infer the physical location of the agent within its surrounding world, specifically, its physical location in a frame of reference. It is usually impossible to directly and accurately deduce one's location from a raw sensor measurement. This is due to two factors: any real sensor measurement will not give full visibility into the every component of the location, and each sensor measurement is often corrupted by errors such as additive white noise or a persistent measurement bias. Given these limitations of sensors, it is important for any localization procedure to be able to combine successive measurements into a consistent belief about the agent and its surrounding world. A belief is a mathematical representation of the knowledge about specified states in the world.

Filtering and estimation algorithms are an answer to the problems stated above. Section 2.2 is dedicated to explaining the types of filtering techniques that are used in this thesis. First, the Bayesian approach to probabilistic estimation is introduced as the framework for all future algorithms used in this thesis. Following that, one of the most pervasive algorithms used in the field, the Kalman filter, is developed in

Section 2.2.2. The Kalman filter is extended to apply to non-linear systems in two ways: linearization around the current state estimate, and a sigma point sampling technique known as the Unscented Kalman Filter (Section 2.2.4). Additionally, its inverse representation, the information filter is presented in Section 2.2.5. Concepts from both of these filters will be used to find solve the optimization problem posed in Section 1.2.1.

In Section 2.3, the different sensors that are used in the localization of the group of agents are enumerated and explained. The particular sensors used in this thesis are: the Global Positioning System (GPS), Inertial Measurement Units (IMUs), Ultra-wideband ranging (UWB), and barometers for altimetry. The principles of their operation are introduced, along with their weaknesses and strengths.

After the basic algorithms for estimation as well as the sensors used in this work have been introduced, selected research based on these concepts in collaborative localization and tracking will be enumerated and discussed. Section 2.4 introduces work in the areas of range-based indoor localization and collaborative algorithms. Work pertaining to optimization for collaborative navigation will also be discussed.

2.2 Probabilistic Estimation

The goal of estimation is to use a set of observed measurements to determine one or more of a system's states of interest with the least amount of error possible. Given that it is usually impossible for an agent to exactly determine states of itself or its environment, it is advantageous to represent one's belief about those states as probability distributions. Using this probabilistic framework, multiple values for a single state and their likelihoods can be considered. If x is a variable representing an state of interest, then the belief of that state can be defined as the following.

$$P(r_0 \leq x \leq r_1) = \int_{r_0}^{r_1} f_X(x)dx \tag{2.1}$$

The function $p_X(x)$ is the probability density function (PDF), which must be positive at every value of x and scaled so that the total area under the function is

one. This probabilistic paradigm for estimation gives the ability to consider multiple and infinite values a given variable, rather than a single best guess.

Given this model for the belief of the state, one can reason about the likelihood that x takes on a certain range of variables, the most probable value of x, the precision of the current best estimate, and many other useful statistics [4]. This thesis will focus on Bayesian probabilistic estimation, where a model of a system and the observations takes the form of a Bayes network, which in many cases can be used to determine the distributions for the system's modeled states.

2.2.1 Recursive Bayesian Estimation

In probability, it can be the case that two events are not independent of each other. In this case, Bayes' Rule gives the relationship between the two random events, A and B (Equation 2.2). The concept can be extended to continuous random variables, and applied to a simple system where there is a known measurement y and a system state x (Equation 2.3).

$$P(A|B) = \frac{P(A \cap B)}{P(B)} = \frac{P(B|A)P(A)}{P(B)} \tag{2.2}$$

$$p_X(x|Y=y) = \frac{p_Y(y|X=x)p_X(x)}{\int_{-\infty}^{\infty} p_Y(y|X=z)p_X(z)dz} \tag{2.3}$$

The function $p_Y(y|X=x)$ is a probability distribution of the measurement, given a certain state x. This likelihood distribution comes from a model of the sensor that produced the measurement. The function $p_X(x)$ is known as the prior distribution of x, and represents any prior knowledge about the variable. The denominator is the marginal probability of seeing the measurement, and is constant for a given measurement y. It serves as a normalization constant for the posterior distribution and is sometimes represented as η.

Almost all real world systems involve more complex inference than the one above, but can still be captured under the model of probabilistic inference. One must con-

sider the case where the system state varies in time, and where more than one measurement is available over time. In such a system, the objective of an estimation algorithm is to determine the probability distribution of the most current state x_n, given measurements $y_{1:n}$ up to and including the present. Bayesian networks provide an efficient way to describe the dependencies between variables in a system, and there exist efficient inference algorithms for systems represented using Bayesian networks. A Bayes network is a directed graph in which every vertex represents a variable in the system. If a vertex representing a variable q has incoming edges from vertices representing variables $r_1, r_2, \ldots, r_n$, then q is dependent on that list of variables, and there exists a function $p_{Q|R_1,R_2,\ldots,R_n}(q|r_1, r_2, \ldots, r_n)$ that models the dependence [26]. One of the simpler types of systems represented by a Bayes network is a dynamic first order Markov system.

If a system is a first order Markov system, its future state solely depends on the current state. In the case of continuous variables, for a given sequence of random variables $X_1, X_2, \ldots, X_n, X_{n+1}$, Equation 2.4 must hold for all n.

$$p_{X_n}(x_n|X_{n-1} = x_{n-1}, \ldots, X_1 = x_1) = p_{X_n}(x_n|X_{n-1} = x_{n-1}) \tag{2.4}$$

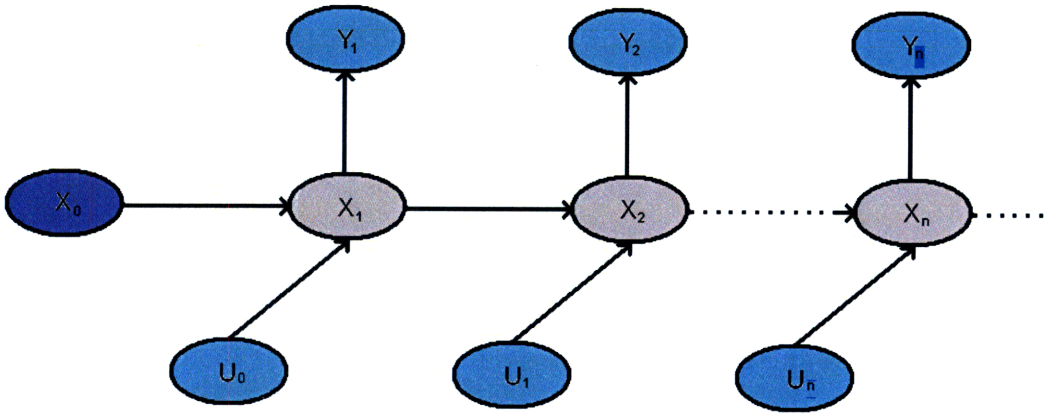

Figure 2-1: A simple Bayes network following the Markov assumption. A distribution for X_0 must be assumed. Marked in light blue are known values such as measured value and deterministic system inputs. Grey are hidden variables, such as an agent's position. The inference problem is to compute distributions of hidden variables given the known values.

In Figure 2-1, x_n represents the current state, which fully represents the system at step n, therefore following the Markov property. x_0 is an assumed initial distribution of the state, and u_n and y_n are two known quantities available at every time step. Another property of the is that y_n is solely dependent on x_n. As this is the case, x_n can be inferred using only the previous state x_{n-1}, previous input u_{n-1}, and the current measurement y_n. The goal for estimation is to reason about the current state given previous measurements, $p_{X_n|Y_{1:n},U_{1:n}}(x_n|y_{1:n}, u_{1:n})$. Given the Bayes network above and a prior distribution for x_0, one can iteratively calculate the PDF for $x_{1:n}$ using Bayesian inference [10].

$$p_{X_n|Y_{1:n},U_{1:n}}(x_n|y_{1:n}, u_{1:n}) = \frac{p_{X_n|Y_{1:n-1},U_{1:n}}(x_n|y_{1:n-1}, u_{1:n}) p_{Y_n|U_{1:t},Y_{1:n-1},X_n}(y_n|u_{1:n}, y_{1:n-1}, x_n)}{p_{Y_n|Y_{1:n-1},U_{1:n}}(y_n|y_{1:n-1}, u_{1:n})} \tag{2.5}$$

The PDF of the current state can be decomposed into three separate PDFs using Bayes' rule. The first PDF in the numerator is the "propagated belief" from x_{n-1} to x_n using the measured value u_{n-1}. The second PDF is the "measurement likelihood", which updates the belief based on the measurement y_{n-1}. The third PDF in the denominator is known as the "evidence".

The propagated belief must be calculated from the last step's posterior belief, $p_{X_{n-1}|Y_{1:n-1},U_{1:n-1}}(x_{n-1}|y_{1:n-1}, u_{1:n-1})$, using the total probability theorem and the state propagation model to marginalize x_{n-1}.

$$p_{X_n|Y_{1:n-1},U_{1:n}}(x_n|y_{1:n-1}, u_{1:n}) = \int p_{X_n|Y_{1:n-1},U_{1:n},X_{n-1}}(x_n|y_{1:n-1}, u_{1:n}, x_{n-1}) p_{X_{n-1}|Y_{1:n-1},U_{1:n-1}}(x_{n-1}|y_{1:n-1}, u_{1:n-1}) dx_{n-1} \tag{2.6}$$

The evidence must then be calculated given the propagated belief in Equation 2.6 and the measurement model by marginalizing x_n.

$$p_{Y_n|Y_{1:n-1},U_{1:n}}(y_n|y_{1:n-1},u_{1:n}) =$$
$$\int p_{Y_n|Y_{1:n-1},U_{1:n},X_n}(y_n|y_{1:n-1},u_{1:n},x_n)p_{X_n|Y_{1:n-1},U_{1:n}}(x_n|y_{1:n-1},u_{1:n})dx_n \quad (2.7)$$

Once the propagated belief and evidence have been calculated, this then allows for the computation of the original Equation 2.5. Noting the simplifications that come from the Markov assumption and the measurement independence from any other variable except for the corresponding state, one can write the iterative solution to the Bayesian network.

Although this estimation method takes into account systems that are dynamic in time and does not constrain the system to a particular family of distributions, several problems in using this type of inference remain for estimation in under real world constraints. Computationally, it is expensive to execute operations on these integrals and functions, except in specific cases, such as those assumed in the Kalman filter.

2.2.2 Kalman Filter

In order to address systems with multiple variables, a single variable x_k is replaced with a vector $\mathbf{x_k}$ of length n. More than one measurement may also be available at a given time, thus y_k becomes vector $\mathbf{y_k}$ of length m. Given multiple variables in these vectors, the previous Bayesian approach of evaluating the integrals and distributions grows exponentially with the number of state variables. Thus, two additional simplifying assumptions can be made to make the problem computationally feasible. Shown in Equations 2.8 through 2.12 is the new system model. The first assumption is linearity, which allows the state propagation and observation models to be described in a matrix formulation (Equations 2.8 and 2.9). The second assumption made is the additive zero mean white Gaussian noise model for the state propagation and observation models. w_k and v_k are vectors of random variables sampled from a zero mean Gaussian PDF with covariances Q_k and R_k, respectively. The Gaussian PDF shown in Equation 2.12 is solely defined by the first and second moments. A_k,

B_k, and C_k are matrices that potentially vary in time.

$$
\begin{aligned}
x_{k+1} &= A_k x_k + B_k u_k + w_k && (2.8)\\
y_k &= C_k x_k + v_k && (2.9)\\
w_k &\sim \mathcal{N}(0, Q_k) && (2.10)\\
v_k &\sim \mathcal{N}(0, R_k) && (2.11)\\
z &\sim \mathcal{N}(\mu, \Sigma) \Rightarrow p_z(z) = \frac{1}{\sqrt{2\pi\Sigma}} e^{\frac{(z-\mu)^T \Sigma^{-1}(z-\mu)}{2}} && (2.12)
\end{aligned}
$$

The belief x_k starts with an initial distribution of $x_0 \sim \mathcal{N}(x_0, P_0)$, and stays Gaussian as it is propagated and updated with a linear model. This allows the belief at any time to be represented solely by a mean $\hat{x_k}$, and a covariance P_k. The Kalman filter mechanizes the process of calculating the mean and covariance of the belief as it is propagated in time and measurements become available [18]. The Kalman filter produces a belief with a mean that minimizes the error $\mathbb{E}[(x - \hat{x})^T(x - \hat{x})]$, and a covariance $\mathbb{E}[(x - \hat{x})(x - \hat{x})^T]$. Furthermore, the computational burden is comparatively light compared to other implementations of the Bayesian filter as the belief is represented as a vector of mean values and a matrix of covariances. All steps for inference can be done as matrix operations on the mean and covariance. Though the Kalman filter requires many assumptions (linear Gaussian Bayesian probabilistic model following the Markov assumption), its applicability is very wide since many the model for many systems can be approximated with it. Though noise may not always be exactly Gaussian, the approximation works well for single mode distributions. The linear assumption may also be foregone under certain circumstances as will be shown in Section 2.2.3.

The Kalman filter's relation to the Bayesian estimation above will be shown along with the algorithm for computing the belief for every instance in time. The first step in Bayesian estimation is to predict the belief of x_k given the observed value of u_k and the previous belief x_{k-1}, previously done by integrating the prior with the propagation model. The belief propagation is simplified to the two following equations.

$$\hat{x}_{k+1|k} = A_k \hat{x}_k + B_k u_k \tag{2.13}$$

$$P_{k+1|k} = A_k P_k A_k^T + Q_k \tag{2.14}$$

The second step is then to calculate the evidence, previously done by integrating the predicted belief with the measurement model. The evidence with mean $\hat{y}_k$ and covariance S_k is now calculated as follows.

$$\hat{y}_k = C_k \hat{x}_{k+1|k} \tag{2.15}$$

$$S_k = C_k P_{k+1|k} C_k^T + R_k \tag{2.16}$$

Given both the predicted belief and the evidence, the posterior can be calculated. This the equivalent to Equation 2.5 in the Bayesian inference section. It should be noted that the update for the mean is a linear addition to the state, according to the previous covariance, the observation matrix, and the measurement evidence.

$$\kappa_k = P_{k+1|k} C_k^T S_k^{-1} \tag{2.17}$$

$$\hat{x}_{k+1|k+1} = \hat{x}_{k+1|k} + \kappa_k [y_k - \hat{y}_k] \tag{2.18}$$

$$P_{k+1|k+1} = [I - \kappa_k C_k] P_{k+1|k} \tag{2.19}$$

In the Kalman filter, $\hat{x}$ and P fully represent the current belief's PDF. $\hat{x}$ is both the mean of the estimate and the most likely value, and P is the covariance of the mean estimate, $\mathbb{E}[(x - \hat{x})(x - \hat{x})^T]$. If the models are accurate and linear, $\mathbb{E}[x - \hat{x}] = 0$. Much of this thesis will focus on bounding the error on the mean belief, $\hat{x}$ by monitoring the estimated covariance P. As the precision of the values of $\hat{x}$ decreases, the corresponding diagonal value representing the variance of that value grows. When a state of $\hat{x}$ is said to be unobservable, the variance of that state will increase without

bound.

2.2.3 Extended Kalman Filter

Many systems of interest are non-linear in nature, and thus cannot directly be modeled using a Kalman filter. The linear assumption can be removed, yielding the following dynamics and observation models, replacing Equations 2.8 and 2.9.

$$x_{k+1} = f(x_k, u_k, w_k) \tag{2.20}$$

$$y_k = h(x_k, v_k) \tag{2.21}$$

$$w_k \sim \mathcal{N}(0, Q_k) \tag{2.22}$$

$$v_k \sim \mathcal{N}(0, R_k) \tag{2.23}$$

Given the above functions, the Jacobians with respect to the inputs can be calculated and used instead of the matrices A, B, C, Q, and R. The method for linearizing around the state estimate is known as the Extended Kalman filter [14].

$$F_k = \frac{df}{dx_k}(x_k, u_k, 0)|_{x_k=\hat{x}_k} \tag{2.24}$$

$$W_k = \frac{df}{dw_k}(x_k, u_k, w_k)|_{x_k=\hat{x}_k} \tag{2.25}$$

$$H_k = \frac{dh}{dx_k}(x_k, 0)|_{x_k=\hat{x}_k} \tag{2.26}$$

$$V_k = \frac{dh}{dv_k}(x_k, v_k)|_{x_k=\hat{x}_k} \tag{2.27}$$

The extended Kalman filtering algorithm is shown below. It is assumed that the Jacobians are calculable via an exterior method, similar to the way f and g are available.

While the Extended Kalman filter performs similarly to the Kalman filter under certain conditions, there are deficiencies due to the non-linearity of the models [30]. The filter is no longer strictly optimal in the least squares sense, and the actual

Algorithm 1 Extended Kalman Filter Algorithm

Require: Prior state belief $\hat{x}_0$, P_0, measured quantities u_{k-1}, y_k, and covariances Q_k, R_k, and functions f, h, $\frac{df}{dx}$, $\frac{df}{dw}$, $\frac{dh}{dx}$, $\frac{dh}{dv}$ $\forall$ times $k = \{1 \dots n\}$,

Ensure: Posterior state belief $\hat{x}_n$, P_n

$k = 1$

while $k \leq n$ **do**

$\hat{x}_{k|k-1} = f(\hat{x}_{k-1|k-1}, u_{k-1}, 0)$

$F_k = \frac{df}{dx_k}(x_k, u_k, 0)|_{x_k=\hat{x}_k}$

$W_k = \frac{df}{dw_k}(x_k, u_k, w_k)|_{x_k=\hat{x}_k}$

$P_{k|k-1} = F_k P_{k-1|k-1} F_k^T + W_k Q_k W_k^T$

$\hat{y}_k = h(\hat{x}_{k|k-1}, 0)$

$H_k = \frac{dh}{dx_k}(x_k, 0)|_{x_k=\hat{x}_{k|k-1}}$

$V_k = \frac{dh}{dv_k}(x_k, v_k)|_{x_k=\hat{x}_{k|k-1}}$

$S_k = H_k P_{k|k-1} H_k^T + V_k R_k V_k^T$

$\kappa_k = P_{k|k-1} H_k^T S_k^{-1}$

$\hat{x}_{k|k} = \hat{x}_{k|k-1} + \kappa_k[y_k - \hat{y}_k]$

$P_{k|k} = [I - \kappa_k H_k] P_{k|k-1}$

$k = k + 1$

end while

return x_n, P_n

belief, if propagated exactly would no longer be strictly Gaussian. Thus, the belief is no longer directly related to the model of the system. Even when the estimated mean is not equal to the true state of the system, the EKF can still operate. However with certain models, it is required to set the initial state to within an ϵ of the true state so that the filter's mean value does not diverge from the truth value. Additionally, if at any time during the filter's execution, the mean goes out of certain bounds from the true state, it can diverge. Even if the belief's mean is correct, the covariance of the error may be understated due to local non-linearities. The next section will show a deterministic sampling strategy to mitigate these effects.

2.2.4 Unscented Kalman Filter

As previously mentioned, it is sometimes the case that the non-linearities of either the state prediction or measurement model are not linear on the order of the current distribution's standard deviation. In the extended Kalman filter, this can cause problems as the models used are linearizations around the mean mean estimate. The

Unscented Kalman filter (UKF) attempts to fix this issue by not only basing the models upon the mean estimate, but also a set of samples around the mean, based on the estimated covariance [17]. All of these samples are propagated through the prediction and measurement models and recombined to form the new mean and covariance estimates.

The first step to implementing the UKF is augmenting the initial state and covariance matrices to include the noise vectors for both prediction and measurement models. As before, $\hat{x}_0 = \mathbb{E}[x_0]$, and $P_0 = \mathbb{E}[(x_0 - \hat{x}_0)(x_0 - \hat{x}_0)^T]$.

$$\hat{x}_0^a = \begin{pmatrix} \hat{x}_0 \\ 0 \\ 0 \end{pmatrix} \tag{2.28}$$

$$P_0^a = \begin{pmatrix} P_0 & 0 & 0 \\ 0 & Q_0 & 0 \\ 0 & 0 & R_0 \end{pmatrix} \tag{2.29}$$

Sample points are then calculated for the prediction and update steps. This deterministic sampling is done by calculating points that are a chosen multiple of the standard deviations from the mean in both the positive and negative directions for every dimension in the augmented state vector. Equations 2.31 through 2.33 are used to generate the sample points. Given tuning parameters α, κ, and β, the weights for calculating the mean and covariance (W^m and W^c) and sample values χ can be calculated. L is the total number of states in the augmented state space.

$$\lambda = \alpha^2(L+\kappa) - L \tag{2.30}$$

$$\chi_0 = \hat{x}^a_{k-1|k-1} \tag{2.31}$$

$$\chi_i = \hat{x}^a_{k-1|k-1} + \sqrt{(L+\lambda)P^a_{k-1|k-1}}_i, \forall i = 1 \ldots L \tag{2.32}$$

$$\chi_{i+L} = \hat{x}^a_{k-1|k-1} - \sqrt{(L+\lambda)P^a_{k-1|k-1}}_i, \forall i = 1 \ldots L \tag{2.33}$$

$$W^m_0 = \frac{\lambda}{L+\lambda} \tag{2.34}$$

$$W^c_0 = \frac{\lambda}{L+\lambda} + 1 - \alpha^2 + \beta \tag{2.35}$$

$$W^m_i = W^c_i = \frac{1}{2(L+\lambda)}, \forall i = 1 \ldots L \tag{2.36}$$

It should be noted that the augmented state vector and its samples χ have three different segments: the original state vector $\chi^x_{k-1|k-1}$, the process noise vector $\chi^w_{k-1|k-1}$, and the measurement noise vector $\chi^v_{k-1|k-1}$. After the sampled points have been calculated, they are all propagated through the prediction equations. The sample mean and covariance can then be retrieved by summing the samples multiplied by their respective weights.

$$\chi^x_{k|k-1} = f(\chi^x_{k-1|k-1}, u_{k-1}, \chi^w_{k-1|k-1}) \tag{2.37}$$

$$\hat{x}_{k|k-1} = \sum_{i=0}^{2L} W^m_i \chi^x_{k|k-1}{}_i \tag{2.38}$$

$$P_{k|k-1} = \sum_{i=0}^{2L} W^c_i (\chi^x_{k|k-1}{}_i - \hat{x}_{k|k-1})(\chi^x_{k|k-1}{}_i - \hat{x}_{k|k-1})^T \tag{2.39}$$

The samples propagated through the prediction equations are also used to generate samples of the predicted measurement distribution. As above, the mean and covariance of the predicted measurement can be calculated by summing over the samples. Additionally, a covariance between the samples in state space and their corresponding samples in the measurement space is calculated.

$$\mathcal{Y} = h(\chi^x_{k-1|k-1}, u_{k-1}, \chi^v_{k-1|k-1}) \tag{2.40}$$

$$\hat{y}_k = \sum_{i=0}^{2L} W_i^m \mathcal{Y}_i \tag{2.41}$$

$$S_k = \sum_{i=0}^{2L} W_i^c (\mathcal{Y}_i - \hat{y}_k)(\mathcal{Y}_i - \hat{y}_k)^T \tag{2.42}$$

$$P_{x_k,y_k} = \sum_{i=0}^{2L} W_i^c (\chi^x_{k|k-1_i} - \hat{x}_{k|k-1})(\mathcal{Y}_i - \hat{y}_k)^T \tag{2.43}$$

Given these quantities, the measurement update is completed by equations similar to those used in the EKF. The Kalman gain is calculated, and the mean and covariance of the belief are updated. Although the UKF acknowledges a non-linear transformation, the end result is still represented by the parameters of a Gaussian distribution.

$$\kappa_k = P_{x_k,y_k} S_k^{-1} \tag{2.44}$$

$$\hat{x}_{k|k} = \hat{x}_{k|k-1} + \kappa_k[y_k - \hat{y}_k] \tag{2.45}$$

$$P_{k|k} = P_{k|k-1} - \kappa_k S_k \kappa_k^T \tag{2.46}$$

2.2.5 The Information Filter

A second form of linear Gaussian estimator is the information filter. While the representation of the belief distribution is different from the Kalman filter, the two filters achieve the same estimated distribution. Instead of using the mean and covariance to represent the Gaussian distribution, an information vector ξ and Fisher information matrix Ω are used, which also fully define a Gaussian distribution. The information matrix is the inverse of the covariance matrix.

$$\xi_k = P_k^{-1}\hat{x}_k \tag{2.47}$$

$$\Omega_k = P_k^{-1} \tag{2.48}$$

Due to the inverse nature of the representation, measurement updates in information form can sometimes be much less computationally intensive. When the number of simultaneous measurements available to a filter are large, it becomes advantageous to switch to the information form in order to incorporate the measurements as sums to the inverted P matrix, as opposed to using Kalman updates. Kalman updates require the computation of the Kalman gain as seen in Line 11 in the Kalman Filter Algorithm. The calculation time for the inversion of S_k is dependent on the number of measurements m, and is lower bounded by $\Omega(m^2 logm)$ [36], whereas the calculation time for the information update is linearly dependent on the number of measurements. Thus, when there is a significantly larger number of measurements than states, the information update shown in 2 should be used instead of Lines 7 to 13 in Algorithm 1. The only additional requirement for this type of update is that the noise of any one sensor reading is not correlated with the noise of any another.

Algorithm 2 Kalman Belief Update using Information Form

Require: Predicted state belief $\hat{x}_{k|k-1}$, $P_{k|k-1}$, measured quantities y_{k_i}, and measurement variances r_{k_i}, and functions h_i, $\frac{dh_i}{dx}$, $\forall$ measurements $i = \{1 \dots m\}$

Ensure: Posterior state belief $\hat{x}_{k|k}$, $P_{k|k}$

$\xi = P_{k|k-1}^{-1}\hat{x}_{k|k-1}$

$\Omega = P_{k|k-1}^{-1}$

for all measurements y_{k_i}, $i = \{1 \dots m\}$ **do**

$H_i = \frac{dh_i}{dx_k}(x_k, 0)|_{x_k=\hat{x}_{k|k-1}}$

$\xi = \xi + H_i^T y_k / r_{k_i}$

$\Omega = \Omega + H_i^T H_i / r_{k_i}$

end for

$P_{k|k} = \Omega^{-1}$

$\hat{x}_{k|k} = P_{k|k}\xi$

return $\hat{x}_{k|k}$, $P_{k|k}$

The state propagation for the information filter is more difficult, and requires the

F_k matrix to be invertible. Similar to the measurement update in the Kalman filter, it is computationally expensive. Shown below are the equations for prediction in the information filter.

$$M_k = F_k^{-T}\Omega_{k-1|k-1}F_k^{-1} \quad (2.49)$$

$$C_k = M_k(M_k + Q_k^{-1})^{-1} \quad (2.50)$$

$$L_k = I - C_k \quad (2.51)$$

$$\Omega_{k|k-1} = L_k M_k^T + C_k Q_k^{-1} C_k^T \quad (2.52)$$

$$\xi_{k|k-1} = L_k F_k^{-T}\xi_{k-1|k-1} \quad (2.53)$$

It can be seen in Equation 2.50 that a matrix as large as the covariance matrix must be inverted. As there is little benefit of using the Information prediction procedure over Kalman filter prediction, it is not used in the rest of this thesis.

2.3 Localization Sensors

Many different types of sensors are available for localization. In this thesis, the algorithm to optimally place sensors around a network is defined. This section will enumerate and describe the sensors used in this experiment, but the effectiveness of the search procedure described is not limited solely to these types of sensors. Many complementary sensor pairs have been used for global localization, the most common of which, in many domains, is the inertial measurement unit (IMU) and global positioning system (GPS) receiver. Section 2.3.1 introduces the IMU and its uses in localization. Section 2.3.3 describes the GPS system components and its uses. In addition to those sensors, ultra-wideband radios for the use of ranging have become available for use in environments where GPS cannot reach [12]. Section 2.3.4 describes these systems and their capabilities.

2.3.1 Inertial Measurements

Inertial measurement units operate by sensing motions of the body to which they are attached. Composed of three gyroscopes and accelerometers, they are able to track position and orientation given initial kinematic information. Inertial sensors range in size from smaller than a few micrometers used in many consumer navigation products to room-sized devices used in submarines during submerged position tracking. The quality of the components vary similarly, with several different grades defined: strategic as the highest, navigation, tactical, and consumer as the lowest. This section will first discuss the components of the IMU and their methods for sensing. The method for determining position using IMUs will then be shown, highlighting the benefits and drawbacks of this sensor type.

In an IMU, three gyroscopes are placed in orthogonal directions to measure the rotation rate vector of the body. Various implementations for rotation sensing rate have been developed. Three of the most widely used technologies will be described. The first gyroscopes were mechanical devices with a spinning mass with high angular momentum which resisted any change in angle [2]. Measurement of the change in angle was made by assuming the spinning mass remained in almost the same orientation as the body moved. These gyroscopes, while incredibly accurate, were still bulky to carry, but have seen a decrease in size.

Another type of gyroscope uses laser light to detect angular rates [7]. This is accomplished by coiling a fiber optic cable around a cylinder. As light passes through the coil, a rotation rate will effectively lengthen the amount of wire through which the light must travel. This is compared to a reference source using an interferometer to detect rotation rates. Fiber optic gyroscopes are smaller and than mechanical gyroscopes, with low error sources. The most novel type of gyroscope has been developed with MEMS technology [1]. Most of these work by vibrating a structure, which will tend to vibrate in the same plane even under rotation. Angular velocity is measured by observing the out of plane motion with respect to the body of the gyroscope due to the Coriolis force. Chip-scale gyros of this type are widely available

generally with poorer accuracy than the previously mentioned gyroscopes.

Similarly to the gyroscopes, three accelerometers are placed in orthogonal directions inside an IMU in order to measure the force vector on the body. The first type of accelerometer was the Pendulous Integrating Gyroscopic Accelerometer (PIGA) [15]. The PIGA measured acceleration via the rotation rate made by the PIGA around the acceleration input axis. Rotations were induced by a gyrating disk with a mass off center from the PIGA's rotating frame. As with the spinning mass gyroscope, this system is highly accurate but bulky. The latest development in accelerometers are MEMS units [37] which measured the movement of a proof mass under restoring forces via piezoelectric components.

Used as dead-reckoning systems, IMUs measure the vector of the force and angular velocity in an inertial coordinate system as previously stated. For the purposes of localization on the Earth, several phenomena in inertial measurements must be accounted for. As the earth frame is rotating in inertial space, sensitive gyroscopes will observe the rotation, which must be subtracted from the rotation in an earth centered, earth fixed frame (ECEF) [32]. Due to the same rotation, the accelerometers will observe a small Coriolis force and a centripetal force in the ECEF, which must be subtracted [33]. The accelerometers will also witness a force of gravity acting on the IMU from the earth's mass. Due to Einstein's equivalence principle, these forces are indistinguishable from true acceleration, and must be subtracted from the readings.

In addition to those forces that come from the difference between the inertial frame and the global frame, the sensors will exhibit imperfect measurements [11]. These come in the form of biases that evolve in time, as well as scale factors that are not one-to-one. Additionally, errors due to non-orthogonality of the sensors and misalignment on the body should be accounted for in models. Additive white noise also appears as a limiting factor for the precision of the instrument. Both accelerometers and gyroscopes are prone to such errors. As accelerometers sense the gravity vector, non-linearities and asymmetries in the sensor output can be seen as well. Since IMUs localize by integrating the rotational velocity and double integrating the acceleration, any error in measurement will be summed, degrading the localization ability of an

IMU alone over time. Shown in Figure 2-2 is the estimation of an agent's location with IMU measurements only. The attitude, position, and velocity were initialized to their exact quantities.

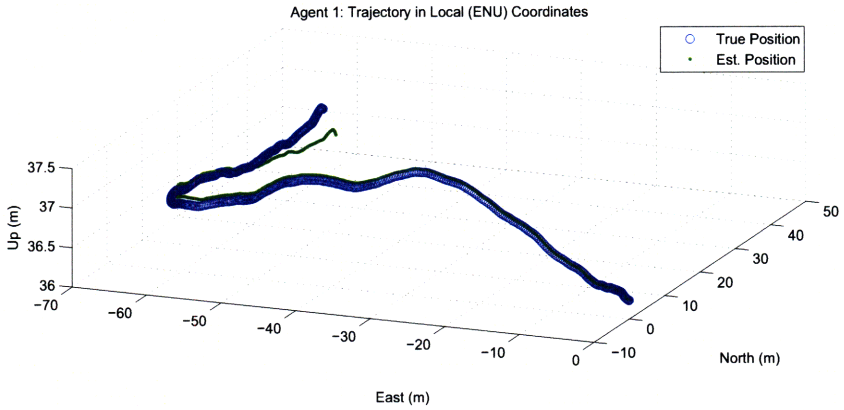

Figure 2-2: Shown in green is an IMU's estimate of the location of an agent as time progresses. Blue denotes the true trajectory. As time progresses, solution is corrupted by IMU errors.

2.3.2 Barometer for Altitude Measurement

Similar to the IMU, many other devices exist for relative positioning information [28]. The baro-altimeter is another such sensor, capable of measuring the difference in altitude from a reference. This sensor measures the pressure exerted by the atmosphere in order to discern the current altitude. A model for the pressure measured by the baro-altimeter can be developed using the following physical relation.

$$p = P_b e^{\frac{-g_0 M(h-h_b)}{RT_b}} \tag{2.54}$$

This equation is based on gravity g acting on an ideal gas with molecular pass M. Reference pressure P_b and temperature T_b at sea level are used to calculate the different in altitude from sea level h_b. Additionally, an ideal gas constant R appears in the equation. Although the pressure of the atmosphere varies depending upon the weather, the effect can be modeled as a slow moving bias that can be estimated [20]. In addition to the bias, errors in the barometer electronics as well as intermittent changes in wind pressure can be modeled as white noise.

2.3.3 Global Positioning System

The second type of sensor used in this thesis' experiments is the Global Positioning System (GPS) [19]. GPS is used for a wide variety of positioning and navigational tasks, from surveying tasks to missile guidance. Three major segments of this system work together to provide position estimates for agents on or near the earth: the control stations, space vehicles, and the receiver.

Control stations serve to monitor the space segment, making any executing maneuvers to maintain the satellite orbits, monitoring the instrumentation on the satellites for errors, regularly synchronizing the atomic clocks of all satellites, and updating their ephemerides and almanacs. The almanac contains course data about all of the satellites' orbits as well as a global model for ionospheric deflections, and is valid for periods of time on the order of days. The ephemeris, a time-invariant parameterized representation of a satellite's precise orbit, is calculated by one of three major control stations and several pre-localized monitoring stations using a Kalman Filter variant. The ephemeris is only valid for up to four hours, and updated by monitoring stations every two hours.

At any given time, 24 or more satellites are operational in the GPS constellation. At least four satellites share a single orbiting plane in the Earth centered inertial coordinate frame, and six separate orbital planes are used. Each orbital plane is separated by 60°, making them equally spaced around the equator. These orbits give line of sight to at least six satellites at virtually any time and any place on the earth's surface. At mid-ranged latitudes, the coverage is often significantly higher. Aboard each satellite is an atomic clock for precise time keeping, a radio receiver for telemetry updates and commands from the control segment, and a radio transmitter to broadcast localization information to the user segment.

Each satellite transmits several signals: the navigation message, the course/acquisition (C/A) code , and the precise (P) code. The navigation data contains time conversion information, the satellite ephemeris data, and the almanac. The C/A and P codes from a single satellite are often referred to as pseudoranges, and contains the precise

time at which the signal was sent from the satellite. Using Code Division Multiple Access (CDMA) technology, satellites transmit this signal on the same carrier frequency, each with its own distinct pseudo-random, but repeating code. C/A codes are transmitted at 1.023 million symbols per second and are 1023 symbols long. Therefore, they repeat every millisecond, and are publicly available. P(Y) codes repeat after a much higher time period, are encrypted, and are not publicly available. Signals generated by the satellites travel at or close to the speed of light in a vacuum to the surface of the earth.

The user segment is possessed by an agent on the ground who wishes to locate themselves. The receiving system consists of a radio receiver, a crystal oscillator or similar quality clock, electronic processing and storage for signal acquisition and localization, and a user or electronic interface. In order to localize, the receiver listens for both the navigation messages and pseudoranges from GPS satellites. Using its internal clock, the receiver generates the same pseudo-random sequences for any given satellite. This generated signal is compared to the received signal via correlation. Given that the user clock will not be synchronized to GPS time in the beginning and will drift at a relatively high rate, an error in the range is induced. Thus, the correlation done with a single satellite is referred to as a pseudorange: a range plus a bias induced by clock synchronization error.

The user can calculate the position of satellites from the ephemeris parameters, and has pseudoranges from those satellites. In order to calculate its three dimensional position, it must also calculate the clock synchronization error, for a total of four unknown quantities. At least four pseudoranges are required solve for these unknown quantities. Various methods for solving for this initial trilateralization problem have been implemented. Given the initial approximate values of the unknowns, an extended Kalman filter can track the position and synchronization error based on future pseudoranges that become available.

Several important factors determine the accuracy of the position fix available to the agent on the ground. The geometry of the satellites with respect to the agent on the ground determines the precision of the position estimate. Shown in Figure

2-3 is a two dimensional representation of how satellite geometry can affect precision. Given that the ranges have a component of white noise, a measurement from both satellites that was within the first standard deviation could have yielded any of the positions in the blue region. For the two satellites coming from almost orthogonal angles, the blue area is small and close to the actual truth. This is less the case with the satellites coming from similar angles.

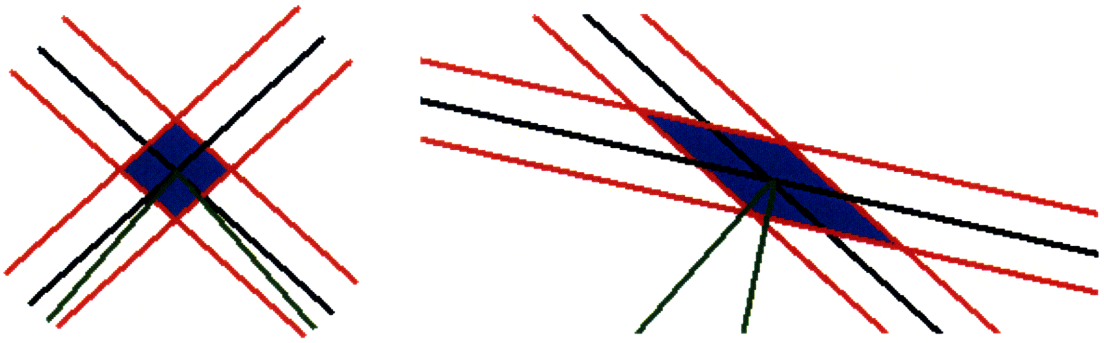

Figure 2-3: Shown are two cases of geometry for two measured ranges from infinite distance. Red indicates the first standard deviation away from the truth range, the black line. The line of sight from the intersection of the two measurements to the signals' origins are shown in green. Blue regions are those within the first standard deviation of both measurements.

A measure of an estimate's accuracy from GPS satellites come from the dilution of precision (DOP) associated with a specific geometry. Geometrical DOP (GDOP) is a figure of merit for the positioning accuracy. The calculation of GDOP is accomplished by forming an information filter with the three position variables in vector x, and the synchronization bias t as states. Initially, the associated information matrix is zeroed, showing no information present. It is then updated with the pseudorange measurements. The inverse of the information matrix is calculated, obtaining the covariance matrix. The square root of the summed diagonals are then used to compute various measures of the positioning and timing accuracy. GDOP is calculated via the position's three diagonal components. A calculated GDOP puts an upper limit on the expected error of localization with the pseudoranges used, assuming the pseudoranges have only additive white Gaussian noise for errors.

Other than errors due to white measurement noise and geometry, several other

phenomena contribute to the errors seen in GPS solution. Due to the change in the electrical permittivity and magnetic permissivity constants associated with the different levels of the atmosphere, the Poynting vector of the signal will be altered from its original direction. This occurs significantly at two interfaces: the exo-ionosphere interface, and strato-troposphere interface. This always adds a positive bias to the range as the wave propagation is no longer along a straight line. Corrections from the satellite almanac mitigate, but do not remove these effects. Additionally, satellite ephemerides and atomic clock synchronization have errors associated with them.

As GPS is quite weak when it reaches the earth surface, many objects in the near field can occlude or reflect GPS signals. Reflected signals will have a positive bias associated with the extra distance traveled, and occluded signals will simply not meet the threshold requirements of a receiver to detect them.

2.3.4 Radio Ranging Measurements

Other than GPS, various types of radio devices may provide ranging capabilities to multi-user groups. Ultra-wideband (UWB) systems serve as a short distance measurement device that works well inside buildings as both a communications as well as for ranging. UWB is defined as any signal that has a bandwidth of at least 20 percent or greater than the base frequency, or a signal that occupies more than 500Mhz of the spectrum [8], in contrast to standard narrow-band signals which typically deviate away from their base carrier frequency by small amounts. One type of UWB signal works by creating a large impulse in the time domain that occupies a large portion of the frequency spectrum. Although the signal occupies many frequencies, it does so at very small power levels, effectively appearing as noise to any narrow-band receivers.

Given the short duration of the pulses, UWB rarely interferes with itself in the presence of multi-path. This is due to the low duty cycle of UWB, and the narrowness of the pulses used. This makes it an ideal candidate for low power communication, and research has been done in the area of using multi-path reflections to improve signal reception [38]. For ranging, UWB's impulse technology also has a great advantage in precision. Due to the narrow pulses in the time domain, correlation yields narrow

peaks, decreasing the uncertainty in the range measurement [3].

In order to obtain a measurement of the distance between any two UWB transceivers without previous time synchronization, a two-way procedure must be initiated. The round-trip procedure begins when the first receiver transmits a signal, making note of the transmit time. The second radio receives the pulse by correlating it against the expected function. It then returns a pulse, noting the amount of time processing time. The first receiver calculates the elapsed time for signal propagation subtracting the processing time on the second receiver as well as other fixed times. Half of the total time of flight back and forth divided by the speed of light yields the distance traveled.

Another type of measurement in an UWB network is derived from receiving a signal at two or more antennas. Time difference of arrival (TDOA) measurements are the difference in arrival of the same signal to several antennas [39], which can be used even if the exact broadcast time of the signal is not known. Given that the distance between two antennas are known and the signal's location is known, the orientation of the two antennas can be calculated. Given two UWB receivers that can find their distance using the time of flight method, pseudoranges received by both units can be used to determine their orientation in a global frame.

In addition to a bound on the accuracy of UWB due to the bandwidth used, several factors in the wave propagation and signal processing can lead to errors in the range measurement. When the transmitter and receiver have line of sight, signal processing errors are dominant, but can normally be represented as additive Gaussian noise, dependent on distance [27]. When the line of sight between transceivers is blocked by material, the model becomes more complicated, as the material can cause dispersion or reflection at the interface. This causes the signal to degrade, on occasion to the point where it is unrecognizable by the receiver. In such cases, a secondary reflection is observed by the receiver and mistaken as the LOS signal, causing a positive bias on the measurement. This bias changes as the line of sight and secondary reflections change with the geometry. Often, this positive bias jumps, and can be detected and quantified with an appropriate model and filter.

2.4 Scenarios for Localization

In the beginning of modern navigational research, most work was done with a single agent attempting to localize and track itself. Various methodologies have been used depending on the situation, from the positioning of celestial bodies to the IMUs used above. As networking technology has become more pervasive, the study has grown to include multiple agent scenarios where agents can communicate with each other. Sharing navigational information has lead to many distributed algorithms for navigation, particularly with UWB communication and ranging. This section will review selected algorithms for localization and optimization that were examined in the process of developing this thesis, and leading up to the ideas presented in the research.

2.4.1 Beacon-based Localization

Although research in time-domain electro-magnetics started in the 1960s, research on its use for range-only measurements for navigation has been a topic of interest. Early in the 1990s, the work started by assuming stationary surveyed beacons for navigation. Feasible means of locating and tracking an agent inside a building were developed using these beacons and a time of flight methodology. Experimental work on the topic has revealed that even within buildings, UWB can be used to locate items and people down to centimeter level accuracy with ranges of up to 100 meters [12]. During these indoor studies, the phenomenon of multipath corruption of range data was discovered to be a leading cause of positioning degradation, mitigated only by steering the antenna such that one of the null regions faced the incoming multipath signal.

To address the issue of multipath, work has been done to use a complementary set of sensors to explicitly estimate the component of range measurements that are due to multipath [16]. In this approach, the two dimensional localization problem was addressed with a gyroscope and odometry used as dead reckoning sensors. UWB receivers provided the necessary ranges for position localization in an externally defined

frame. A probabilistic model of UWB multipath was generated from gathered data, and it was found that the multipath component normally stayed at its previous value or made a significant jump to another value. Given this non-Gaussian behavior, a particle filter was used for estimating the beacon bias, demonstrating significantly better performance than using EKF techniques with additional mechanization to detect and reject outlier measurements.

Research has been done to remove the requirement that beacons must be located prior to using them to locate an agent. In such studies known as Simultaneous Localization and Mapping (SLAM), beacons are deployed and considered stationary, but their exact location is unknown. One approach looks at estimating the location of beacons as well as the agent through the use of an EKF that accepts beacon to beacon as well as beacon to agent measurements, and two additional techniques that allow for better localization [9]. The first technique is the use of an optimization technique to improve the map, minimizing the innovation of the ranging measurements as well as the error of the beacon locations. The second technique uses the odometry information to create pseudo-beacons, increasing the number of connected beacons and the chance that the map is rigid enough to localize all beacons. A similar approach uses geometrical knowledge to ascertain the subset of measured ranges that agree with each other. This information is then used to reject outliers based on that geometrical knowledge [24]. With these SLAM techniques, there exists a greater flexibility in missions that can be accomplished with human agents as beacons can be placed in the environment during missions. However, this still may not be an option in military scenarios where unguarded instrumentation cannot be left behind.

Other research has taken on the task of removing the constraint of requiring beacons to be placed before the scenario. This was meant to allow unknown environments to be explored with a group of robots sharing localization information. It was shown that by ensuring that certain robots in a collection remained motionless while others moved, localization information could be held for a long duration using trilateration from those temporary beacons [22]. In a local coordinate frame defined by the first set of temporary beacons, agents explore and localize by "leapfrogging" each other,

intermittently exchanging roles of exploration and stationary beacon. Using an EKF, ultrasound ranging along with odometry are fused to obtain positions of the exploring robots. When switching roles to a stationary beacon, the corresponding states are removed from the Kalman filter and localization of the new beacon with respect to the old ones is calculated using a non-linear optimization technique. Although this technique works for robots, such dynamic constraints are usually not applicable to localization involving human agents, who may not hold a position for a localization algorithm.

2.4.2 Dead Reckoning Systems

In addition to the work with ranging devices, research has been done on bounding the error growth in GPS-denied environment using dead reckoning systems. The goal of these systems is to use information about the kinematics of the agent to track their location without the reliance on any external agents or objects that would not be available prior to a mission. The purpose of these systems is to maintain localization information until external measurements like GPS can be reacquired or the mission is over.

While an IMU is a simple solution to this problem, there are other additional aiding devices that can help an agent retain positioning information for longer. Given information about the kinematics of a system, an IMU can be calibrated to remove some of the larger components of errors that build up. One idea presented is to place an IMU on an agent's shoe. At each footfall the IMU detects, it can use the stationary position of the foot to do a zero velocity update (ZUPT) [23]. Knowing that the body frame is not moving allows for the estimation of the biases on the accelerometers and gyroscopes. Given that this can be done at every footfall, errors can now be measured as a percentage of the distance traveled instead of time elapsed, similar to odometry. In addition to the ZUPTs, research suggests that a range measurement between a person's two feet in addition to an IMU on each foot would further reduce the error accumulated during GPS outages [5]. Although these systems decrease the error accrued over time, they would still benefit from collaboratively sharing information

with other agents.

2.4.3 Optimization Problems in Localization

As agents share information with each other, the filters that handle the information become more complex, adding extra states for each agent, and in the case of inter-agent range measurements, a fully centralized filter is required to handle a number of measurements that scales quadratically with the number of agents. Given the numerous sensor readings, it may not be computationally feasible to use all of them. Given a bound on computational ability, only the most useful measurements should be incorporated. Research into an optimization to determine the measurements to incorporate resulted in the formulation of the following optimization problem [21].

$$
\begin{aligned}
\textbf{minimize} \quad & trace(W_p P_{ss} W_p^T) \\
\textbf{subject to} \quad & F_c P_{ss} + P_{ss} F_c^T + Q_c - P_{ss} C P_{ss} = 0 \\
& C = \sum_{i=1}^{M} f_i H_i R_i^{-1} H_i^T \\
& 0 \leq f_i \leq f_{i_{max}}, i = 1 \ldots M \\
& \sum_{i=1}^{M} f_i \leq f_{total} \\
& e_{3i}^T P_{ss} e_{3i} \leq \epsilon_\phi, i = 1 \ldots N
\end{aligned}
$$

Figure 2-4: Localization Optimization under Computational Constraints. The sum of diagonals of a weighted covariance matrix is minimized while bounding the number of fused measurements per unit time.

A weighting of the steady state covariance matrix P_{ss} from the Ricatti equation is minimized, with measurement information coming into the Ricatti equation through the C matrix. The C matrix is a sum of all measurement information, which is upper bounded by both the maximum rate for each sensor $f_{i_{max}}$ and the maximum rate for incorporating measurements into the filter f_{total}. The final constraint limits the error on the state associated with current attitude to ensure small linearization errors. The most severe limitation of this optimization problem is that it involves the steady state

Ricatti equation, which requires that the measurement and state prediction matrices do not change in time. When inter-agent ranging is an available measurement, this requires that the agents remain in a static formation for the optimization to be valid.

The formulation of the optimization problem presented in this thesis was inspired from the optimization problem in Figure 2-4. As seen in the next chapter, many of the same equations appear. Similar to the optimization problem above, the problem of deciding about what measurements to include in the filtering strategy is addressed. However, significant changes are made to the formulation, and the solution strategy is different. Additionally, the optimization problem posed in this thesis benefits from being based on the propagation of the covariance matrix instead of steady state Ricatti Equation.

Chapter 3

Optimization Definition and Solution Strategy

3.1 Chapter Overview

This chapter will address the problem presented in the beginning of the thesis, using the concepts of estimation and localization presented in Chapter 2. The goal is to instrument a group of agents with a subset of predefined sensors in such a way that they can track themselves in a given environment while following a specific trajectory, minimizing the cost of the entire configuration. Section 3.2 will define the optimization problem, including the system model, the optimization search space, the covariance constraints, and the objective function. Although a short list of sensors is considered in this thesis, the optimization will be presented in a general way that allows for the consideration of various sensor types and configurations with mixed sensor types. Section 3.3 outlines the solution strategy for the optimization problem, including a description of structure of the problem that allows for the implicit elimination of instrumentations without directly testing them. As the solution strategy requires the calculation of the covariance at every timestep under varying sensor configurations, the algorithm for propagating the covariance in an efficient manner is described in Section 3.4. Several experimental search strategies for sensor configurations are shown in Section 3.5.

This optimization problem and its solution are used to solve the specific problem of how to instrument a group of inter-ranging agents with varying grades of IMUs available to them under varying environments and mission types. These environments lead to degraded GPS measurements due to near-field obstructions of the GPS satellite signals in a way that would be seen during missions in rural settings, cities, and indoor environments. In each section explaining the optimization, the details of each specific problem will be addressed after the general optimization definition.

3.2 Optimization Definition

Presented in Figure 3-1 is the formal statement of the optimization problem that this work solves. In this general formulation, the configuration C is an ordered set of sensor assignments s_j. A sensor assignment can take on integer values from 0, which represents the lowest quality sensor available or no sensor at all, to M_j, the maximum quality sensor available. The function f_{cost} takes a configuration as an argument and produces a scalar quantity representing the cost of the configuration. Given a sensor configuration C, the `system_model` function produces a set of matrices for propagating the covariance $P_{k|k}$, as seen in Section 2.2.3, at every time k. The covariance on the error at the first time is used to initialize P and a bound, γ, on the traces of multiple weighted covariance matrices is imposed at every timestep.

In the specific problem addressed in this thesis, a group of agents with limited GPS are sharing measurements as well as ranging between each other. These agents must be instrumented with IMUs so as to allow for satisfactory localization in a global frame. Two different qualities of IMUs are available to be placed one per agent, as well as the option of having no IMU on any agent. Given that there are N agents and 2 qualities of IMUs, $M_j = 2$ and constraint 3.2 becomes $s_j \in \{0, 1, 2\}, \forall j = 1 \ldots N$.

The localization of every agent in the network is the goal in the IMU assignment problem. Thus, the bound in constraint 3.8 applies to the variance of the position states for all agents in the network separately. For every agent, a weighting matrix W_{p_i} serves to isolate only the covariance entries associated with the position of

$$
\begin{aligned}
\textbf{minimize} \quad & f_{cost}(C = \langle s_1, s_2, \ldots, s_L \rangle) && (3.1)\\
\textbf{subject to} \quad & s_j \in \{0, 1, 2, \ldots, M_j\}, \forall j = 1 \ldots L && (3.2)\\
& F_k, Q_k, H_k, R_k \leftarrow \texttt{system\_model}(C, k), \forall k = 1 \ldots T && (3.3)\\
& P_{0|0} = \mathbb{E}[\tilde{x}_0 \tilde{x}_0^T] && (3.4)\\
& P_{k|k-1} = F_{k-1} P_{k-1|k-1} F_{k-1}^T + Q_{k-1}, \forall k = 1 \ldots T && (3.5)\\
& S_k = H_k P_{k|k-1} H_k^T + R_k, \forall k = 1 \ldots T && (3.6)\\
& P_{k|k} = [I - P_{k|k-1} H_k^T S_k^{-1} H_k] P_{k|k-1}, \forall k = 1 \ldots T && (3.7)\\
& \texttt{Tr}(W_i P_{k|k} W_i^T) < \gamma, \forall k = 1 \ldots T, \forall i = 1 \ldots N && (3.8)
\end{aligned}
$$

Figure 3-1: Sensor instrumentation optimization with covariance constraints. The cost of the instrumentation is minimized while enforcing bounds on covariance. The system model which includes environmental factors and agent trajectories along with a sensor configuration determine the quality and availability of measurements.

that user, which is bounded by a constant value $loc^2_{max-err}$. The constraint becomes $\texttt{Tr}(W_{p_i} P_{k|k} W_{p_i}^T) < loc^2_{max-err}$.

Given the intermittent nature of the pseudorange measurements, IMUs may have to be placed on agents so that the entire group can endure the loss of complete GPS or poor GDOP for lengths of time. Each IMU has an intrinsic cost of being placed in the network. This is a representation of actual price of purchasing and deploying the IMU package as opposed to GPS receivers and UWB transceivers alone. The specific objective function for IMU placement is seen in Equation 3.9, and uses the representative cost of the IMUs. The cost of the low quality IMU is approximately the cost of a tactical grade IMU, and the cost of the high quality IMU is the cost of a navigation grade IMU.

$$
f_{cost}(C = \langle s_1, s_2, \ldots, s_N \rangle) = \sum_{j=1}^{N} = \begin{cases} 0 & \text{if } s_j = 0 \text{ No IMU (NO)} \\ 500 & \text{if } s_j = 1 \text{ low quality IMU (LQ)} \\ 10,000 & \text{if } s_j = 2 \text{ high quality IMU (HQ)} \end{cases} \tag{3.9}
$$

The generation of F_k, Q_k, H_k, and R_k matrices by the system model for the IMU placement problem are broken up into two sections: the environment definition,

and trajectory definition. These two components along with measurement models presented in Section 3.4 fully define the system model.

3.2.1 Environment Definition

The system model determines the availability of measurements in the H_k and R_k matrices, be they from the baro-altimeter, GPS, or UWB inter-agent ranging. In this work, the availability of GPS pseudoranges is dependent on the existence of line of sight (LOS) to the satellite from which it originated. A GPS receiver is assumed to be available to every agent in the network, but pseudoranges are subject to blockages in the environment due to buildings as well as the earth's surface. An environmental map is defined for an optimization problem, which contains (1) a list of buildings, their positions, and their dimensions, and (2) the orbital parameters for each GPS satellite.

Defined in a local East/North/Up (ENU) coordinate system, the environment map is used to calculate when line of sight to a satellite is obscured. This happens when the user is inside a building, or outside when the LOS intersects a plane of a building model. The calculation of the availability of pseudo-ranges is discussed further in the trajectory generation section (4.2.4). UWB availability is based on the relative positions of the agents and a maximum UWB range. The UWB maximum range is a function of the density of buildings in the environment. Depending on the test case, baro-altimeter readings are either available to all agents every second or to none. The secondary purpose of the environment is to serve as a way of generating trajectories for users in the environment using waypoints and their connecting edges as a graph of feasible paths for the optimization problem.

Although in this optimization problem the H_k and R_k measurement matrices come from the environment model with no dependence on the IMU instrumentation, the measurement model can vary based on the instrumentation available. Described in Section 3.3.6 is a method for solving the optimization problem where measurement devices other than IMUs are being considered for the network.

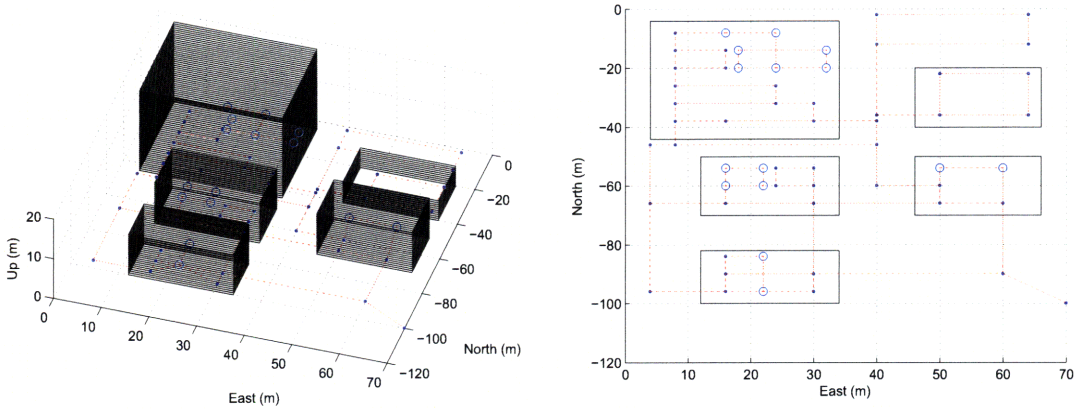

Figure 3-2: Example environment map in 3-dimensional space (left) and from above (right). Buildings indicated with black with way points and feasible paths through them in red. Waypoints on the paths are blue.

3.2.2 Trajectory Definition

For every optimization problem, trajectories for each of the agents are defined as a list of body locations in three dimensional space and body orientations with three degrees of freedom. These two agent states are defined at every time interval for a given time period, and are known as the truth values. This is where the agent actually is in the trajectory, free of corruptions of measurements or errors from filtering. Given the true positions and attitudes at every time, the true rotation rates, velocities, and accelerations can be calculated. As the agents go along their trajectories and satellites orbit the earth, realistic varying geometries are seen with respect to inter-user ranging as well as the available pseudoranges. These are used to generate the expected measurements for the optimization problem, which affect the F_k, Q_k, H_k and R_k matrices at every time k. The generation of trajectories is further discussed in Section 4.2.

The solution to the sensor optimization depends on the specific trajectories of the agents because the inter-agent ranging measurements as well as the GPS pseudorange measurements have varying affects on localization accuracy. As shown in Section 2.3.3, the localization accuracy of ranging systems depends largely on the geometry of the agents involved. Additionally, the effects of IMU propagation on the components of localization error produces more geometrical affects.

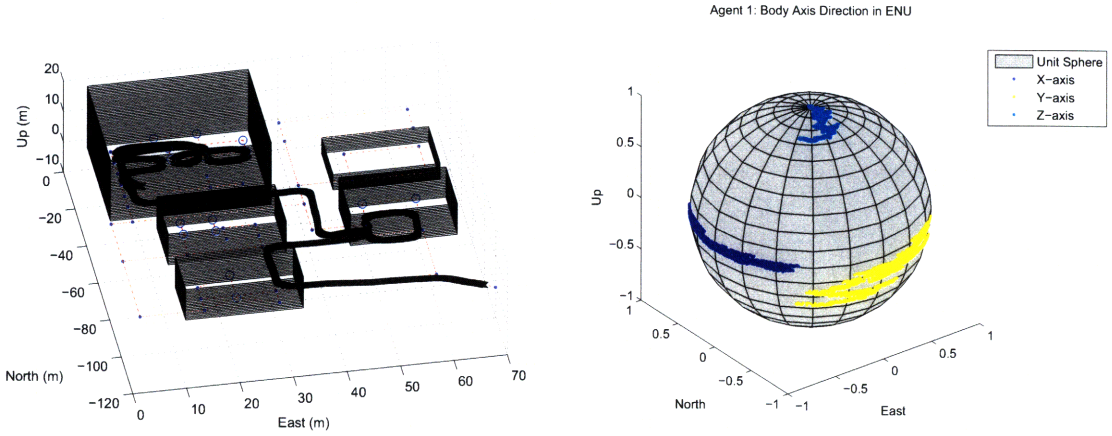

Figure 3-3: Example agent trajectory position and attitude. The agent's progression along a path can be seen imposed on a map (left). The body orientation with respect to the local East/North/Up coordinate system is also shown (right).

3.2.3 Search Space and Objective Function

For this optimization problem, all possible instrumentations of the network of agents comprises the space over which the optimization routine must search. In constraint 3.2, it can be seen that every sensor in the configuration s_i has a total of $M_i + 1$ assignments. Any combination of sensor assignments is considered a valid configuration; thus, there are $\prod_{i=1}^{N} M_i + 1$ possible instrumentations for the optimization in the general case. In the case of the IMU assignment problem, three values for s_i are allowed. N users are in the group, and any combination of the IMU assignments in the cluster are allowed. This describes the search space for the problem, which contains 3^N possible instrumentations.

The objective function ranks the instrumentations according to their expense. The only requirement for the objective function is to be able to produce a value for every instrumentation, which should be based on their relative desirability. Given this ordering, the objective of the optimization is to determine the lowest ranking configuration in terms of cost that meets the covariance propagation and localization constraints. The objective function in this optimization sums the cost of the IMUs to obtain the instrumentation's total cost. In this case, using kinematics to propagate an agent's states costs nothing, a cheap IMU has low cost approximate to a tactical

grade device, and a high quality IMU has a higher cost associated with a navigation grade device. Given this, the permutation of the positions of IMUs of the same quality around the network does not affect the cost, creating many configurations with equal costs but different localization accuracies.

3.2.4 Covariance Propagation and Constraints

Four of the constraints in the optimization problem are present to enforce the covariance results that would be seen when a Kalman filter is executed. The initial covariance is defined using the covariance of the error of the average starting value for the state, and is enforced by constraint 3.4. Using the Kalman filter equations for predicting and updating the covariance in Algorithm 1, the relationship between the covariance at time step $k-1$ and k is defined in the optimization as constraints 3.5 through 3.7. This relationship depends upon the models for measurements and propagation, as well as the environment which determines whether measurements are available. For the optimization, the truth value for the state is used to generate the Jacobians for covariance propagation.

In addition to the constraints that show the relationship between covariance matrices in a Kalman filter there is a bound on the values in the covariance matrix as seen in constraint 3.8 that enforces a limit on the expected error of the position estimate at all times. A weighting matrix W_i pulls out the entries of the matrix associated with the covariance of agent i's position error. The trace examines the sum of the diagonals of this matrix, each of which represents the variance of a component of the position error. After estimation, linear Kalman filtering asserts that the error of a state $\tilde{x}_n$ is a random variable that exhibits a zero mean Gaussian probability distribution with variance $P_{n,n}$. The localization metric $\sqrt{P_x + P_y + P_z}$ chosen is similar to GDOP, which is used to measure the localization quality of GPS measurements alone as outlined in 2.3.3. Putting a lower bound on the trace of the position covariance at every time step also limits the expected error in localization, just as with GDOP.

A requirement on the system model is that if a instrumentation assignment $s_i = j$ yields a value of γ_j at time k for the left hand side of constraint 3.8, then an assign-

ment only changing s_i to be less than j must make the left hand side of the constraint produce a value greater or equal to γ_j for all time. Informally stated for the IMU assignment problem, a cheaper IMU instead of a more expensive one must produce a worse estimate of position at all times. This requirement can easily be met when considering different grades of IMUs for the users, and makes the optimization problem feasible to solve in a limited amount of time. The way to ensure this requirement on the system model will be further discussed in Section 3.3.2.

3.2.5 Optimal Solution

A solution to this optimization problem is defined as the instrumentation assignment for which there exists no strictly lower cost instrumentation such that the constraints are met. Therefore, any optimization routine that wishes to find an optimal solution must be able to prove that all instrumentations that cost less than the provided answer do not meet the optimization's constraints. There may be more than one optimal solution, since there can exist configurations that have the same cost and provide the desired accuracy. Thus, it is possible to have multiple optimal solutions.

3.3 Optimization Solution Algorithm

This section details the approach to solving this optimization problem, starting with an analysis of a straight-forward but computationally infeasible approach for any non-trivial problem size. Two key points about the relationships between instrumentations in the optimization problem are introduced. Using these relationships, a procedure for eliminating large numbers of instrumentations as the solution to the optimization will be introduced.

3.3.1 Brute Force Approach

A simple approach to solving this optimization problem is to iterate through a list of all possible instrumentations while keeping track of the lowest cost instrumentation

that has met the constraints. When the every configuration has been exhausted, an optimum instrumentation is found, if it exists. For each instrumentation, the cost is evaluated. If the cost is lower than the current best known instrumentation, the constraints are imposed by evaluating constraint 3.3 is used to obtain the matrices for covariance propagation. Starting with $P_{0|0}$, the predicting and updating equations are evaluated. After each update, the covariance bound constraint 3.8 is checked. If constraint 3.8 is violated at any time, the current configuration cannot be a solution to the optimization problem. However, if the ends of all trajectories is reached without any constraint violations, the configuration is the new lowest cost configuration.

Algorithm 3 Brute Force Solution to Instrumentation Optimization

Require: cost function $f_{cost}(C = \langle s_1, s_2, \ldots, s_L \rangle)$,
system model $[F_k, Q_k, H_k, R_k] = \texttt{system\_model}(C, k)$,
initial covariance $P_{0|0}$
Ensure: $\nexists\ C$ such that $cost(C^*) > cost(C)$ and C^* and C meet all optimization constraints
$C^* = \emptyset$
$cost_{min} = \infty$
for all instrumentations C in search space **do**
  $P = P_{0|0}$
  for all $k = 1 \ldots T$ **do**
    $[F_{k-1}, Q_{k-1}, H_{k-1}, R_{k-1}] = \texttt{system\_model}(C, k-1)$
    $P_{k|k-1} = F_{k-1} P_{k-1} F_{k-1}^T + Q_{k-1}$
    $P_{k|k} = [I - P_{k|k-1} H_k^T (H_k P_{k|k-1} H_k^T + R_k)^{-1} H_k] P_{k|k-1}$
    for all $i = 1 \ldots N$ **do**
      if $\texttt{Tr}(W_{p_i} P_{k|k} W_{p_i}^T) \geq \gamma$ **then**
        skip to next instrumentation
      end if
    end for
    if $f_{cost}(C) < cost_{min}$ **then**
      $C^* = C$
      $cost_{min} = f_{cost}(C)$
    end if
  end for
end for
return C^*

Each covariance propagation for an instrumentation requires time on the order of the length of the trajectory, and is dependent on the number of agents. When

the number of agents is large, fusing the inter-range measurements becomes the most time consuming step, as the number of measurements grows quadratically with agents. Given that an inversion of a square matrix with the same size as number of measurements must be executed, this puts the covariance propagation's lower bound on time complexity at $\Omega(n^4 log(n))$. As the covariance propagation potentially must be done for every instrumentation, the time to find the solution becomes is the multiplication of the time required for covariance propagation and the number of instrumentations. Several observations about the problem can be made to reduce the time required to solve the problem, making it feasible for significant numbers of agents.

3.3.2 Performance Relation Definition

The first observation that can be made about the problem is that performing the covariance propagation test on a single instrumentation to ensure the constraints are satisfied can potentially imply the outcome of many other instrumentations. Second, a relation between the instrumentations can be defined to show that some are inferior to others. An instrumentation A is said to be inferior to another instrumentation B when the maximum value of the performance metric $\mathtt{Tr}(W_{p_i} P_{k|k} W_{p_i}^T)$ over all agents $i = 1 \ldots N$ and all time $k = 1 \ldots T$, for A is strictly greater than B's maximum value. As constraint 3.8 must hold for all agents over all time, if instrumentation B fails to meet the constraint, instrumentation A would also fail to meet the constraint.

In the optimization problem solved in this thesis, some instrumentations can be shown to be inferior to others. The following Figure 3-4 is an example of where this relation can be seen experimentally during simulation. It is shown that we can compute this inferiority relationship between two sensor configurations without simulation, but based on the properties of the system. Three agents are assigned IMUs of high quality $C_1 = \{HQ, HQ, HQ\}$, versus a configuration where 2 agents are assigned the high quality IMU and a single agent has a low quality IMU $C_2 = \{HQ, HQ, LQ\}$. All other measurement availabilities are the same. As the performance metrics never even cross each other, the maximum of C_2 is greater than C_1, thus C_2 is inferior to C_1. In the system model that is used, this is assured by a requirement on the system

that the possible sensors for a single assignment have an order of strict superiority.

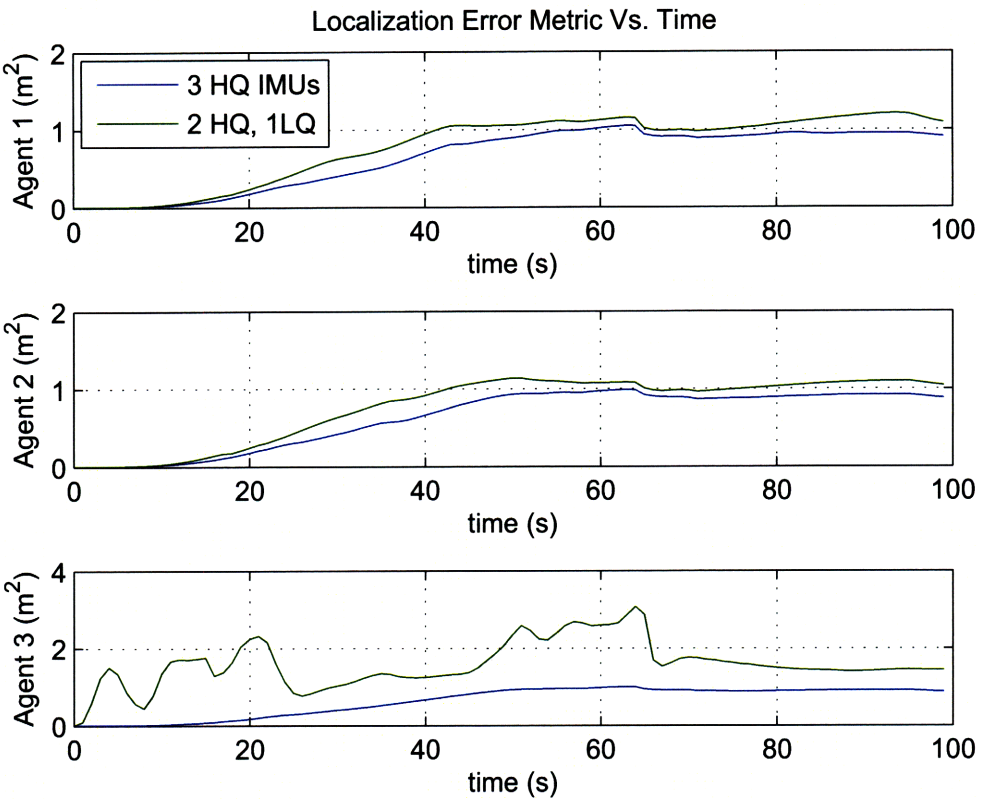

Figure 3-4: Instrumentation Performance Relation Example. The three high quality IMU configuration's performance metric is always less than the two high quality and one low quality IMU configuration.

The strict superiority system requirement on measurement models ensures that sensors being considered for a single assignment s_i are ordered by quality, each assignment j being superior or equal to the next lowest assignment $j-1$. In general, a measurement model A is superior or equal to another B if one or more of the following properties holds for every time step:

1. for every diagonal entry in process noise covariance matrix Q_A and Q_B, A's entry is less than or equal to B's corresponding entry, or

2. for every diagonal entry in measurement noise covariance matrix R_A and R_B, A's entry is less than or equal to B's corresponding entry, or

3. H_A and R_A contain at least all rows present in H_B and R_B. (Zero or more additional measurements are available in measurement model A.)

In each of these cases, the diagonals of P_A are guaranteed to be less than or equal to the corresponding diagonals of P_B for all time. If P_A violates the covariance bounding constraint, P_B will certainly as well.

The concept of inferior sensors and configurations in the IMU problem can be shown mathematically by the diagonal components in the Q_k matrix. The diagonals that are used for the low quality IMU are larger than those used for the high quality IMU due to the noisier sensors. Given that the same update measurements are available to both configurations, the trace of the covariance matrix will always be larger for instrumentation C_2. This relation holds for many other possible configurations for the group of agents. Any configuration $C_q = \{s_1, s_2, \ldots s_N\}$ is inferior to a given configuration $C_r = \{s_1, s_2, \ldots s_N\}$ if for all sensors in C_r, the corresponding sensor assignment in C_q is less than or equal to the sensor in C_r, and there exists at least one IMU in C_q that is strictly less than its corresponding IMU in C_r.

A directed acyclic graph (DAG) showing the relation of all instrumentations can be developed for the optimization problem. A graph consists of a list of vertices and edges. Vertices are visualized as objects, and edges are lines connecting those objects. Directed acyclic graphs have edges that only go in a single direction. They also have the property that no path exists from a vertex back to itself. As the inferiority relation is transitive, a path from any vertex to another in Figure 3-5's graph denotes an inferiority relationship between the two represented configurations. This graph is generated by enumerating through every configuration, linking them to all configurations for which one sensor is inferior and the other assignments are equal. Although there may be more inferiority relations as defined in this section, the relationships described above can be solely determined by the configurations themselves without running the entire covariance propagation. This allows the potential to address multiple configurations without iterating through their respective covariances through time. If a test configuration with multiple descendants in the DAG violates the covariance constraints, the result of the descendant's covariance test is implied to

violate the covariance constraints as well. The exact number of configurations that are inferior to a given configuration can be calculated via the expansion of all its descendants in the DAG.

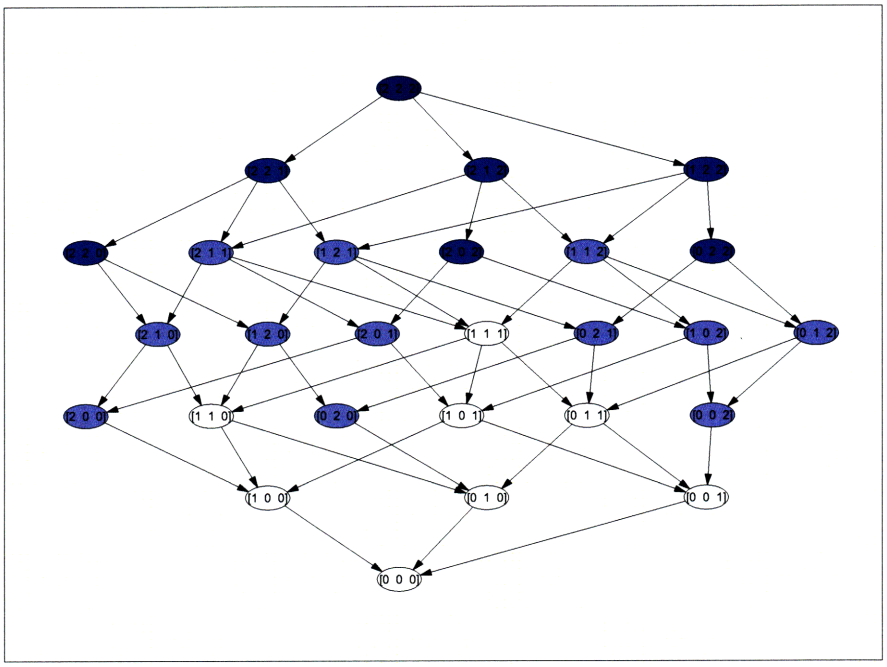

Figure 3-5: Instrumentation Performance Relation Graph. Three agents have a choice between IMU configurations 1, 2, and 3. Shade of blue indicates cost.

3.3.3 Cost Relation Definition

A relation between the costs of the instrumentations within the search space can also be used. It is possible to calculate the cost of a configuration without propagating the covariance forward in time by examining the cost of each component in the configuration. Given this information, a list of costs with the respective number of configurations at that cost can be created by iterating through all configurations and placing them in bins by cost. The number of configurations that are more expensive than a given configuration can be calculated by examining the list of costs.

3.3.4 Elimination of Configurations

Given the cost and performance relations defined above, the testing of a configuration for meeting the covariance constraints reveals more information about the outcome of the testing of other configurations.

1. If a given configuration fails to meet the covariance constraints, none of the configurations that are inferior will meet those covariance constraints either. Thus, those configurations need not be tested.

2. If a configuration meets the covariance constraints, it is unnecessary to test any configuration that costs more, as the minimum cost configuration is what must be found.

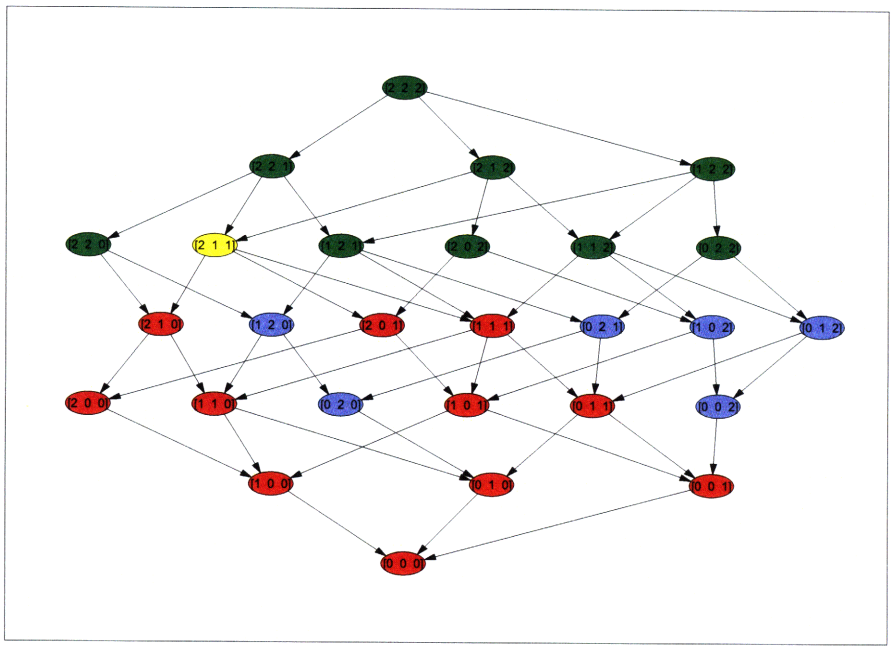

Figure 3-6: Graph of instrumentations affected by the test of the instrumentation represented in yellow. Green vertices are eliminated if the configuration passes, red vertices are eliminated if the configuration fails. Blue vertices are unaffected.

The objective is to minimize the amount of time required to test instrumentations to find an optimal configuration. Given that regardless of the test outcome, multiple

configurations can be eliminated, it is desirable to choose the configuration that will implicitly eliminate the most other configurations once tested. This approach of eliminating the most configurations at each step is not necessarily optimal in minimizing the number of instrumentations that must be tested. Instead, it is a greedy strategy that examines the current state of the search problem with previously eliminated configurations, and chooses the next configuration based on that state.

3.3.5 Search Procedure

Instead of iterating through all configurations as the brute force approach does, this search procedure updates a data structure based on the outcome of a configuration test, has a procedure for querying that data structure to calculate the next best configuration to test, but still produces an optimal configuration as the brute force method does. Based on the strategy for choosing the test configurations, the data structure and its space complexity changes. Section 3.5 considers multiple strategies, comparing the times required to find the optimal configuration. Algorithm 4 shows the generic structure that all of the algorithms follow.

Algorithm 4 High Level Search Procedure for Optimal Solution

Require: cost function $f_{cost}(C = \langle s_1, s_2, \ldots, s_L \rangle)$,
system model $[F_k, Q_k, H_k, R_k] = \texttt{system\_model}(C, k)$,
initial covariance $P_{0|0}$
Ensure: $\nexists\ C$ such that $cost(C^*) > cost(C)$ and C^* and C meet all optimization constraints
$C^* = \emptyset$
$S = \texttt{create\_configuration\_structure}()$
$C = \texttt{query\_configuration\_structure}(S)$
while $C \neq \emptyset$ **do**
  $R = \texttt{test\_configuration}(C, \texttt{system\_model}, P_{0|0})$
  $S = \texttt{update\_configuration\_structure}(S, C, R)$
  if $R = passed$ AND $f_{cost}(C) < f_{cost}(C^*)$ **then**
    $C^* = C$
  end if
  $C = \texttt{query\_configuration\_structure}(S)$
end while
return C^*

The variable S denotes the structure that is used to store the state of the configurations. The information that is stored in this structure varies by the method for choosing configurations to test (Section 3.5), and is created by the subfunction `create_configuration_structure`. The `query_configuration_structure`(S) function computes the next configuration to test for meeting the optimization constraints given the outcomes of previously tested configurations. The function `update_configuration_structure`(S, C, R) updates the data structure to include the result R of the covariance test on configuration C. C^* holds the best configuration found so far. The `test_configuration` function iterates through the time steps, testing the covariance matrices at every time step for a specific sensor configuration to see if it meets the covariance bound. This covariance propagation procedure is discussed further in Section 3.4.

3.3.6 System Model Requirements

The optimization and search strategy has been presented side by side with the example of assigning one IMU per agent in a group following trajectories. However, many other sensor choosing problems can be solved using these exact methods. Another example of a problem solvable with this method is the instrumentation of a single user with a variety of sensors in order to meet a localization requirement. In such a scenario, a single user may have the option of incorporating measurements from a GPS receiver, angle or range to beacons, inertial, magnetometer, odometry, or a baro-altimeter, all of varying qualities. The algorithm presented can be used to find the sensor suite of minimum cost that meets localization requirements. For direction-oriented sensors such as cameras, this optimization framework can be used to choose between a variety of angles for varying localization requirements.

In order to be solved using this method, the system with the available sensors must have a few properties. The noise on the sensors must be able to be approximated by a Gaussian probability distribution, and the Jacobian matrix for both the measurement and state propagation models must be calculable. Within the maximum localization error bound, there must not be a significant difference in the Jacobian

produced from the linearization around the current estimated mean. An estimator accepting stochastic measurements will almost never have the correct mean for linearization. In order for the Jacobians produced during the covariance analysis that is done in the optimization to match what a stochastic filter's Jacobians would be, the linearization must not be sensitive to deviations within the maximum localization bound. Additional covariance bounds can be introduced to enforce this requirement.

3.4 Covariance Propagation

The optimization problem presented in Figure 3-1 has three constraints (3.5, 3.6, 3.7) for the propagation of covariance through every time step of the trajectory, along with an additional constraint specifying the prior belief of the state (3.4). Additional constraints (3.8) require that the covariance matrix have bounds at every time step. To enforce these constraints, the solution presented in this thesis implements the `test_configuration`$(C, \texttt{system\_model}, P_{0|0})$ function, which tests a configuration C with a given system model and original covariance $P_{0|0}$.

Given the covariance propagation and update matrices F, Q, H, and R, the equations can simply be evaluated. Presented in this section is the state space layout of the IMU assignment problem and the specific ways in which the matrices are generated. Mathematical measurement models are introduced for all sensors used in the problem. In addition, equivalent and faster ways of computing the covariance are introduced using both the Kalman and Information filter paradigms.

3.4.1 State Space Setup and Initial Covariance

The state space contains the system states that must be estimated to localize the agents. The state space is setup in two sections: the per agent states, and the network global states. The first section includes the kinematics states and error states of any instrumentation the user caries. In the model used for the IMU placement problem, the following is a list of the per agent states.

1. position (p): the current 3 dimensional position of the agent in the earth centered, earth fixed (ECEF)reference frame. This frame is cartesian with the center at the middle of the earth, the x-axis pointing towards the Prime Meridian, the z-axis pointing towards the north pole, and the y-axis pointing in the direction that makes the system orthogonal and right-handed. This coordinate frame rotates with the earth.

2. velocity (v): the current three dimensional velocity of the agent in ECEF.

3. attitude (ϕ): the three small angles of rotation along the axes in the ECEF frame.

4. accelerometer time evolving bias (b_{accel}): the additive biases on the three accelerometers that change with time.

5. gyroscope time evolving bias (b_{gyro}): the additive biases on the three gyroscopes that change with time.

In total, there are 15 states for every agent present in the network. In addition, there are three states that apply to the network as a whole.

1. clock bias (b_{clk}): the additive bias by which the shared network time is different from the GPS time.

2. block drift (d_{clk}): the drift rate of the clock bias for the shared network time.

3. baroaltimeter bias (b_{alt}): the additive bias that the altimeters collectively see due to local changes in atmospheric pressure.

The initial covariance for the kinematic states was set to zero so that the localization bound would not be exceeded. As the process noise is introduced into the system, the transient low covariance values for the kinematic states reach their realistic values. The initial values for the IMU, GPS, and baroaltimeter error states is discussed with their models.

3.4.2 State Prediction and Measurement Update

State prediction is done using the equations provided by the Kalman filter. This requires the current belief about the state to be in covariance form. The prediction equations are simple matrix multiplication and addition, as presented in Section 2.2.3. However, in the implementation of the covariance propagation for the IMU assignment problem, the number of measurements grows quadratically with respect to the number of states. This is due to the inter-range measurements between all of the agents. For this reason, instead of propagating the covariance with a Kalman update, Algorithm 2, which is based on the information filter is used. Although the covariance matrix must be inverted twice, the measurements can be fused by addition and multiplication with no need to invert the of the predicted measurement innovation covariance S_k.

3.4.3 Error States as Gauss-Markov Processes

For most error states in the models used, the error model used is a Gauss-Markov process [6] where values have a normal distribution. The equation for this process has two parameters: a stability σ, which is the standard deviation for the values of the process over all time, and a time constant τ which determines the length of the process's exponential correlation in time. The state propagation for this type of Markov process m is shown in Equation 3.11. Covariance propagation is also shown (Equation 3.12). In this thesis, this model is used for the clock drift, IMU bias errors, and baro-altimeter bias.

$$\lambda = e^{\frac{-dt}{\tau}} \tag{3.10}$$

$$m^+ = \lambda m + \sqrt{1-\lambda^2} N(0, \sigma^2) \tag{3.11}$$

$$P^+ = \lambda^2 P + (1-\lambda^2)\sigma \tag{3.12}$$

3.4.4 State Propagation using the IMU

A high quality model of an IMU can normally have upwards of six states per accelerometer and gyro. A permanent as well as time varying bias for both scale factor and bias account for four states. Misalignment of the sensor with respect to the body account for another two states, for a total of more than 36 states for a single IMU. However, for this thesis, two different types of errors were examined for gyros and accelerometers: white noise and a time correlated bias. The permanent biases and misalignments can be previously determined, and scale factor errors will not dominate given the dynamics of human agents, which is the primary mode considered in this thesis. For accelerometers and gyroscopes, the sensor models in Equations 3.13 and 3.14 are assumed, respectively. The variable b_{accel} is a time correlated Gauss-Markov process with parameters st_{accel} and τ_{accel}. The variable b_{gyro} is a time correlated Gauss-Markov process with parameters st_{gyro} and τ_{gyro}.

$$a = a_{bdy} + b_{accel} + N(0, \sigma^2_{accel} dt) \tag{3.13}$$

$$g = \omega_{bdy} + b_{gyro} + N(0, \sigma^2_{gyro} dt) \tag{3.14}$$

The variable a_{bdy} is the acceleration undergone by the accelerometer in its axis of sensitivity. Similarly, the variable ω_{bdy} is the angular velocity undergone by the gyroscope in its axis of sensitivity. Two different IMU models are used: IMU_1 has the capabilities of a low quality tactical grade IMU, and IMU_2 has the capabilities of a high quality navigation grade IMU.

It should be noted that the noise on all of the Gauss-Markov processes of accelerometers and gyroscopes, as well as the white noise on each sensor, in IMU_1 are strictly greater than the noise in IMU_2. As the estimation of velocity and position are in the ECEF frame, several effects appear due to earth rotation. The Coriolis F_{cor} and centrifugal F_{cent} forces appear in the accelerometers, and and an additive rate ω_E appears in the gyroscopes. In addition, gravity due to the earth is also present in the measurements shown in the accelerometers. These effects must be accounted for

Table 3.1: IMU Measurement Model Parameter Values

Parameter	Value	Unit
st_{accel1}	3.3000×10^{-3}	g
τ_{accel1}	2.5000×10^{-1}	hr
st_{gyro1}	1.0000×10^{2}	$\frac{^\circ}{hr}$
τ_{gyro1}	2.5000×10^{-1}	hr
σ_{accel1}	3.2400×10^{1}	$\frac{m}{s}/\sqrt{hr}$
σ_{gyro1}	1.0000×10^{0}	$^\circ/\sqrt{hr}$
st_{accel2}	2.4000×10^{-4}	g
τ_{accel2}	2.5000×10^{-1}	hr
st_{gyro2}	2.4000×10^{-1}	$\frac{^\circ}{hr}$
τ_{gyro2}	2.5000×10^{-1}	hr
σ_{accel2}	2.0000×10^{-3}	$\frac{m}{s}/\sqrt{hr}$
σ_{gyro2}	3.0000×10^{-3}	$^\circ/\sqrt{hr}$

in the system model.

$$\omega_E = 0.250697 \frac{^\circ}{s} \hat{z} \tag{3.15}$$

$$\omega_{bdy} = (\omega_{ECEF} + \omega_E)dt \tag{3.16}$$

$$F_{cor} = -2\omega_E \times v_k \tag{3.17}$$

$$F_{cent} = \omega_E \times (\omega_E \times p_k) \tag{3.18}$$

$$F_{grav} = -p_k/|p_k|^3 \tag{3.19}$$

$$a_{bdy} = (a_{ECEF} + F_{cor} + F_{cent} + F_{grav})dt \tag{3.20}$$

In order to obtain positioning information from the IMU measurements, they must be used to propagate the current belief about the kinematics in a process known as strapdown navigation [32] [33]. The body rotation rate ω_k from the gyroscopes with sensor errors and earth rate removed is used to update the attitude as stored in the directional cosine matrix A. $(\omega_k\times)$ represents the 3-by-3 skew symmetric matrix formed by the three elements of the ω_k. The attitude and the acceleration a is then used to update the velocity v. In turn, the velocity updates the position p.

$$A_k = (I - (\omega_k \times))A_{k-1} \tag{3.21}$$

$$v_k = A_{k-1} a_k dt + v_{k-1} \tag{3.22}$$

$$p_k = v_k dt + p_{k-1} \tag{3.23}$$

3.4.5 Kinematic State Propagation

Without IMU measurements, state propagation is done via the same kinematics equations, with the sensor models used by the IMUs. Instead of receiving measurements, zero values for body angular rate as well as body acceleration are given to the filter for state propagation. Given this, the terms for all inertial forces in the ECEF frame and gravity, as well as earth rotation are removed from the kinematic equations. To account for the difference between the actual accelerations and rotation rates that an agent will undergo, the white noise on the body rates are increased. The expected value of velocity change is set to 1 $\frac{m}{s}$ per second, and the expected value of angle change is set to 1 radian per second.

Table 3.2: Kinematic State Model Parameter Values

Parameter	Value	Unit
st_{accel3}	3.3000×10^{-3}	g
τ_{accel3}	2.5000×10^{-1}	hr
st_{gyro3}	1.0000×10^{2}	$\frac{^\circ}{hr}$
τ_{gyro3}	2.5000×10^{-1}	hr
σ_{accel3}	6.0000×10^{1}	$\frac{m}{s}/\sqrt{hr}$
σ_{gyro3}	6.0000×10^{1}	$^\circ/\sqrt{hr}$

3.4.6 Inter-user Ranging Model

When the range between two agents in the network is available, the model assumed in this thesis is that there is a white noise component with standard deviation σ_r, and no bias component. This simplified model ignores the issue of multipath which occurs in urban and indoor environments frequently. Overcoming the problems presented

by multipath is a current research topic that has been addressed with a range of techniques from mitigation using multiple antennas [29] to estimation with particle filters [27]. For each agent pair with position p^a and p^b respectively, a measurement is based on their ECEF positions as shown in Equation 3.24.

$$h_{baro} = \sqrt{(p_x^a - p_x^b)^2 + (p_y^a - p_y^b)^2 + (p_z^a - p_z^b)^2} + N(0, \sigma_r^2) \tag{3.24}$$

For each matrix row in the Jacobian, range-based measurements is generated by a placing a unit vector facing in the opposite direction to the source in the elements corresponding to the position of the agent. In the case of inter-user ranging, one row will contain two unit vectors for each agent involved, pointing away from the other agent.

3.4.7 GPS Pseudorange Model

When pseudo-ranges are available to an agent in the network, they share a common bias due to the network time drifting from GPS time. In addition, there is a white noise component with standard deviation σ_{pr} that accounts for phenomena such as a deviations of the atmospheric model, ephemeris errors, and clock errors of the satellites. For each agent and satellite pair with position p and s respectively, a measurement is based on their ECEF positions as shown in Equation 3.25.

$$h_{baro} = \sqrt{(p_x - s_x)^2 + (p_y - s_y)^2 + (p_z - s_z)^2} + b_{clk} + N(0, \sigma_{pr}^2) \tag{3.25}$$

$$b_{clk}^+ = b_{clk} + d_{clk}dt + N(0, \sigma_{clk}^2) \tag{3.26}$$

The clock bias is determined by the integration of the clock drift and the addition of white noise with standard deviation σ_{clk} in the integration (Equation 3.26). The clock drift is modeled as a continuous Gauss-Markov process with a time constant τ_{dr} and stability st_{dr}. The noise on the pseudorange measurement is large enough to

take into account ephemeris and residual atmospheric errors after correction by an atmospheric model.

3.4.8 Altimeter Model

When baro-altimeters are used in IMU assignment problem, they are on every agent in the network, and measurements are available at 1 Hz intervals. Every baro-altimeter in the network shares a common bias due to the current atmospheric pressure in the region, but with an uncorrelated Gaussian white noise that represents local pressure phenomena and sensor noise. For each agent, a measurement is based on their ECEF position components as shown in Equation 3.27.

$$h_{baro} = \sqrt{p_x^2 + p_y^2 + p_z^2} - R_E + b_{alt} + N(0, \sigma_{alt}^2) \tag{3.27}$$

R_E is the radius of the earth, and σ_{alt} is the standard deviation of the baro-altimeter noise. The state b_{alt} is modeled as a continuous Gauss-Markov process with time constant τ_{alt} and stability st_{alt}.

Table 3.3: Ranging Measurement Model Parameter Values

Parameter	Value	Unit
σ_r	5.0000×10^{-2}	m
σ_{pr}	3.0000×10^{0}	m
dt	1.0000×10^{-1}	s
σ_{clk}	3.0000×10^{-1}	m
τ_{dr}	9.0000×10^{2}	s
st_{dr}	3.0000×10^{-1}	$\frac{m}{s}$
R_E	6.3781×10^{6}	m
σ_{alt}	7.5000×10^{-1}	m
τ_{alt}	6.0000×10^{0}	hr
st_{alt}	5.0000×10^{0}	m

3.5 Selecting Test Configurations

Testing the covariance of a particular instrumentation is a time consuming task, especially for long trajectories and large numbers of states. The time-saving component

of the algorithm to solve the optimization problem is to choose instrumentations for covariance testing in an intelligent way so as to eliminate the most instrumentations for every covariance test. The approach outlined in Algorithm 4 is greedy as it only makes the decision based on the number of eliminations that can be made with the next covariance test. Two main strategies for choosing instrumentations were tested: maximizing the expected number of instrumentations that are implicitly eliminated over the two possible results of the covariance test, and maximizing the minimum number of eliminations over the two possible results.

These two techniques are similar to those used to find the best data point to use in order to train a classifier. In [35], binary classification of news articles is the goal. In order to do this, training data must be selected, but classifying data points for training is considered to be slow; thus, in order to yield the maximum benefit for training, the algorithm selects data points to classify based on the previous data points and their classifications. The space of classifiers that successfully explains all of the training data can be split by each successive classification of training data. The structure of this optimization problem is similar in that there exists a number of configurations that must be classified as passing the covariance constraints or not passing them. The classification of a single configuration takes a considerable amount of time. Given all of the configurations, it is desired to test the one configuration that classifies the most other configurations. Within these two general strategies (maximizing the minimum number of classified configurations and maximizing the expected number of classified configurations), different methods for calculating the number of eliminations are tested for computational time requirements.

3.5.1 Maximizing Expected Eliminations

To maximize the number of expected eliminations, one must iterate through all of the instrumentations, calculating the number of implicit eliminations upon an instrumentation both meeting the covariance constraints and upon not meeting the covariance constraints. If it is assumed that the likelihood of passing and failing the covariance test is equal, the objective becomes to maximize $\frac{n_{failed}+n_{passed}}{2}$ over all configurations

(max-expected). Although other assumptions could be made about the likelihood of various instrumentations passing the covariance test, equal probability is used.

In order to calculate the number of implicit eliminations that would occur upon knowing the outcome of the covariance test, two data structures were used. A directed acyclic graph is used to represent the performance relations of the configurations as seen in Figure 3-7. In order to calculate the number of implicit eliminations if the configuration fails to meet the covariance constraints, the number of vertices can be counted during a search for all vertices reachable from the test configuration's vertex. The second data structure that is used is a lookup table that holds the number of configurations that would be eliminated by testing a configuration of a certain cost. Evaluating the cost function using the test configuration in conjunction with this lookup table, the number of configurations that would be eliminated upon showing that the covariance bounds are met is known.

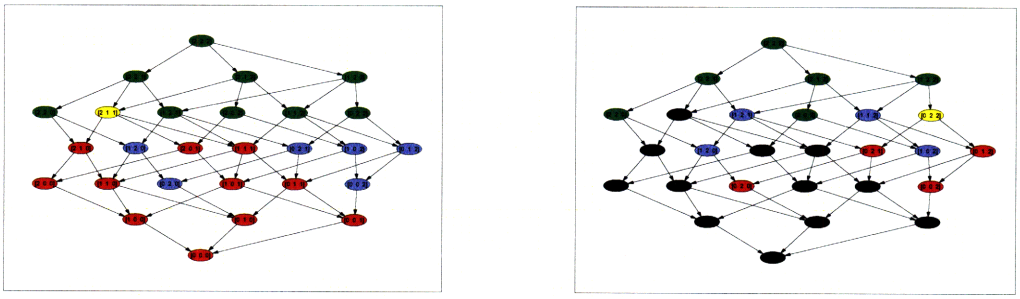

Figure 3-7: Performance Relation Directed Acyclic Graphs. Shows two consecutive choices for covariance tests when there are three agents. Black vertices have been eliminated by the previous test, and thus should not be counted in any further calculations of eliminated configurations.

The DAG used to compute the number of inferior configurations has a vertex for every configuration in the search space. As the edges are stored in a matrix, the total space required is on the order of $O(M^{2N})$, where M is the number of sensors available for a given assignment, and N is the number of sensors that must be assigned. This first approach was the most intuitive way to calculate number of eliminated configurations for a given test configuration, as well as keeping track of previously eliminated configurations.

3.5.2 Maximizing Minimum Eliminations

The second approach to finding the best configuration to test was to maximize the minimum number of configurations that would be eliminated given that the configuration would be tested (max-min). Under the two possible outcomes of each covariance test, this algorithm guarantees that the configuration with the most number of implicit eliminations under the worst cast outcome is tested. Additionally, if given the same worst case outcome, the algorithm prefers the configuration that yields the greater number of eliminations in the alternate case.

Several different methods for calculating the configuration that maximized the minimum number of eliminations are used. The first method is exactly as was used for the maximum expectation structure. However, for sensor configurations with large numbers of sensor assignments, the data structure becomes too large. A second method that only requires a data structure that holds the previously tested configurations as well as their outcomes is introduced.

3.5.3 Iteration Method

Given two configurations and no extra data structure, it is possible to compute whether one would be eliminated by the other if tested. Given configuration $C_A = \langle s_1, s_2, \ldots, s_L \rangle$ and C_B, if every sensor in C_A is of equal or greater quality to the corresponding sensor assignment in C_B, then C_B is eliminated if C_A fails the covariance test. If the evaluation of $f_{cost}(C_B)$ is greater than $f_{cost}(C_A)$ and C_A passes the covariance test, then C_B is eliminated.

Given that the number of eliminations can be calculated in this way, the best configuration to test for meeting the covariance constraints can be determined through iteration, as seen in Algorithm 5. An outer loop goes through all configurations in the search space, searching for the best configuration to test. An inner loop goes through all configurations in the search space, comparing them to the prospective test configuration, determining whether that configuration would be eliminated under a passed or failed covariance test. This method for searching only requires a data

structure as large as the number of tested configurations.

Although the data structure required is much smaller, the time required to run is $O(M^{2N})$ as the two loops through the entire instrumentation space are nested.

3.5.4 Sampling Method

In order to reduce the run time of the iteration method, an additional modification was made. Instead of iterating through all configurations, a uniform sample of all configurations that have not been previously eliminated was desired. In Algorithm 5, both the outer and inner loops through all configurations are replaced with uniform sampling of the search space. Although this method does not calculate the best configuration to test according to the max-min approach, the algorithm has comparable results in terms of the number of configurations that are tested, and reduces the total amount of time in problems with more numbers of configurations.

An issue of the sampling method is that as more configurations are tested, the search space decreases. Iterating through or sampling only the configurations left in the search space is non-trivial, as it requires that the solution be less than a given cost and also not inferior to any previous configurations shown to have failed. Sampling the entire search space and rejecting the previously eliminated samples becomes less feasible as the remaining search space diminishes. In order to counter this problem, a single table of values that correspond to each configuration is kept to pick only configurations that have not been eliminated. This structure requires a space on the order of M^N, and the search requires time on the order of M^N as it must search this structure.

3.5.5 Runtime Comparisons

In the IMU placement problem, as the number of agents in the network increases, the search space increases. A comparison of the various methods described are applied to this problem while varying the number of agents to show how the performance is affected. Figure 3-8 shows both the total required time to decide which configura-

Algorithm 5 Test Configuration Search Under Iteration Method

Require: cost function $f_{cost}(C = \langle s_1, s_2, \ldots, s_L \rangle)$,
previous test outcomes P
Ensure: next configuration to test C
$C = \emptyset$
$max\_min = 0$
for all instrumentations $test\_conf$ **do**
for all previous test outcomes $prev$ in P **do**
if $test\_conf$ inferior or equal to $prev : conf$ AND $prev : result = failed$ OR
$test\_conf$ costs more or same as $prev : conf$ AND $prev : result = passed$
then
skip $test\_conf$
end if
end for
$pass\_elim = 0$
$fail\_elim = 0$
for all instrumentations $elim\_conf$ **do**
for all previous test outcomes $prev$ in P **do**
if $elim\_conf$ inferior or equal to $prev : conf$ AND $prev : result = failed$
OR
$elim\_conf$ costs more or same as $prev : conf$ AND $prev : result = passed$
then
skip $elim\_conf$
end if
end for
if $elim\_conf$ inferior or equal to $test\_conf$ **then**
$fail\_elim++$
end if
if $elim\_conf$ costs more or same as $test\_conf$ **then**
$pass\_elim = 0$
end if
end for
if $max\_min > pass\_elim$ AND $max\_min > fail\_elim$ **then**
$C = test\_conf$
if $pass\_elim > fail\_elim$ **then**
$max\_min = pass\_elim$
else
$max\_min = fail\_elim$
end if
end if
end for
return C

tion to test, as well as the total time spent testing the covariance constraints. Four methods for choosing test configurations are shown. "max-min sampled" uses the random sampling method described in Section 3.5.4 and "max-min iteration" uses the method described in Section 3.5.3. "max-min structure" uses the DAG and lookup table described in Sections 3.3.2 and 3.3.3 to find the configuration that maximizes the minimum number of eliminated configurations. "max-expectation" uses the same data structures, but finds the configuration that maximizes the expected number of eliminated configurations, assuming an equal probability of violating or meeting the covariance constraints.

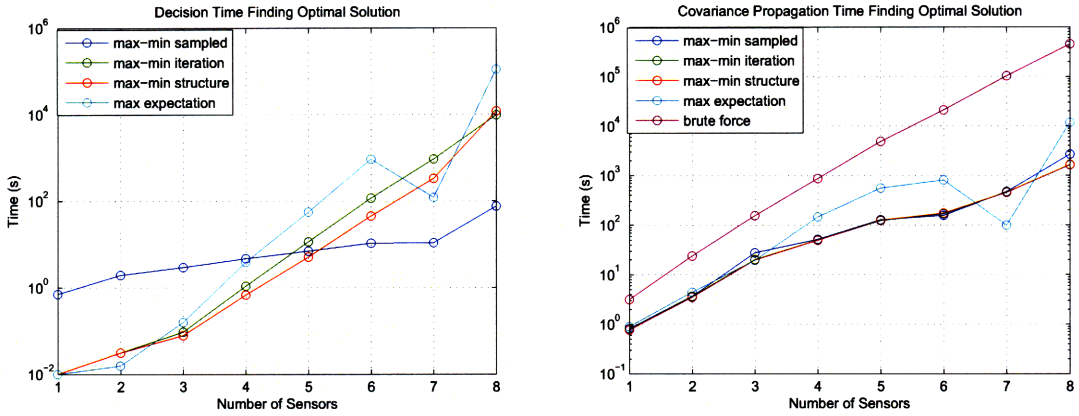

Figure 3-8: Instrumentation selection time and covariance propagation time as a function of the number of users.

Maximizing the expected number of eliminations proved to scale the worst of all of the algorithms. Often, it chose the highest cost configuration or a configuration that was inferior to all other configurations. As the algorithm attempts to maximize $\frac{n_{failed}+n_{passed}}{2}$, these extremes appear attractive to test. In these cases, one of n_{failed} or n_{passed} represents almost all of the configurations, and the other contains only one configuration. For most interesting problems, the configuration that minimizes the cost and meets all of the constraints will not be an extreme. Thus, the configurations that max-expected test will almost always yield the result for which only one configuration is eliminated. The one data point for which max-expected performed better than max-min occurred because the solution was one of the lowest cost configurations, which max-expected tested first. Among the max-min procedures,

the sampling procedure can be seen to follow a different trend. The data structure method executes a graph search for each configuration, and the iteration method has nested loops that traverse the entire search space, whereas the sampling procedure traverses the search space once in order to find configurations that have not been eliminated only once. However, its overhead at lower numbers of sensors is due to the number of samples taken regardless of problem size. It can also be seen that the max-min procedures do not differ much from each other in terms of the amount of time required for covariance propagation, showing that while the max-min sampling algorithm may pick sub-optimal configurations for test, it does not affect the amount of time spend propagating covariance for configurations.

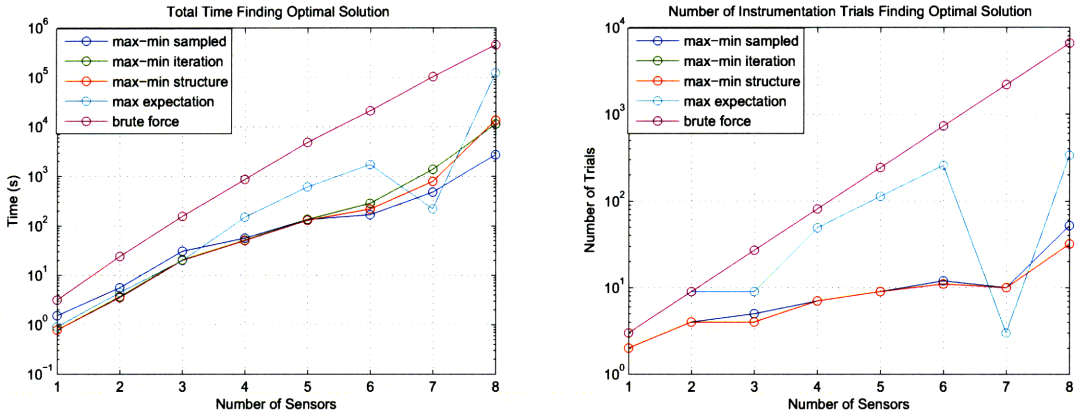

Figure 3-9: Total time to solve optimization problem, as well as number of instrumentations tested.

The number of trials between the max-min algorithms vary slightly with the sampling algorithm requiring a few more covariance trials. This shows that the effectiveness of the algorithms in choosing good configurations is similar. In the total amount of time required, the sampling procedure is shown to scale best with the number of IMU assignments that must be made, but is on the same order of all of the max-min techniques. All of the max-min techniques take approximately two orders of magnitude less time at six agents than the brute force solution presented in Algorithm 3.

THIS PAGE INTENTIONALLY LEFT BLANK

Chapter 4

Optimization Verification and Environment Testing

4.1 Chapter Overview

Using the optimization solution techniques presented in Chapter 3, the solutions to IMU assignment under varying conditions is examined. As stated, the IMU assignment problem requires that trajectories are available for each agent in order to account for geometry in the ranging measurements, as well as the direction of the IMU components. The first section of this chapter details the way in which trajectories, as well as derivative values are calculated. These values include kinematics used for IMU measurements and kinematic propagation, and the availability of pseudoranges and inter-agent range measurements.

Section 4.3 introduces the method used in this thesis to verify the validity of solutions produced by the optimization search procedure. Given a set of representative test trajectories, measurements are generated according to the models assumed by the optimization problem. These measurements are then fused using a Kalman filter implementation. The end goal of these procedures is to ensure that the covariance results of the optimization procedure matchare consistent with that of a Kalman filter's covariance. Additionally, the error in the Kalman filter mean localization estimate with respect to the truth data is shown to be consistent with the covariance statistics.

After the search algorithm's results are shown to match the results of the Kalman filter accepting stochastic measurements, the optimization is used to investigate the effect of changing various parameters about the environment, the trajectories, and the optimization itself. Additionally, three main environment types are defined and tested: indoor, urban, and suburban.

4.2 Trajectory and Truth Data Generation

In order to formulate the optimization problem for IMU placement, trajectories for every agent must be generated. A requirement of the trajectories is to be realistic in terms of what would be seen for a person conducting either a military or emergency response mission. A trajectory can be minimally defined as a list of attitudes and positions at set intervals in time, and is generated from a previously constructed map. From the minimally defined trajectory, velocity, acceleration, and angular velocity are generated. In addition to the kinematics, the availability of GPS pseudoranges is calculated using the trajectories and buildings in the map. The parameters used to generate the trajectories are shown, and the procedures for generating both the maps and trajectories are outlined.

4.2.1 Map Generation

In order to generate trajectories for the agents in the IMU placement problem, a map is used. Associated with a map is a GPS location of the center of the map, a list of buildings, the ephemerides of all GPS satellites, and a graph of waypoints and feasible paths. A list of vertices with associated waypoint locations is stored in a graph. The edges of the graph are stored in a boolean matrix, which determines whether a route between the two waypoints is feasible. The map serves two purposes: a framework from which trajectories are generated, and to determine the availability of pseudoranges to the agents as a function of the space vehicle and time.

Maps are generated via scaling multiple predesigned structures. These structures are then shifted into different coordinates in the local frame and connected together

Table 4.1: Map Generation Parameters

Parameter	Indoor Value	Urban Value	Suburban Value	Unit
Tile Scale	10	10	20	m
Number of Tiles	16	25	25	#
Mean Building Height	25	25	10	m
Std. Dev. Building Height	10	10	5	m
Track Coverage	90%	60%	30%	%

in a tiling procedure. The structures are manipulated via three parameters: building height, track coverage, and tile type. Building height determines the height of the tallest building in the tile, track coverage determines the percentage of the track is covered by buildings, and tile type determines the arrangement of the buildings and the connections of the feasible paths on the tile. The tiles are designed such that the entire graph in the map stays completely connected. In order to mimic different environments, both the size of the tiles as well as the tile parameters are changed to yield a map with a certain ground coverage, an average building height, and varying building types. Three different map types are examined in this thesis: a suburban environment, an urban environment, and a mostly indoor environment. Each environment is generated using different parameters for the tiles.

4.2.2 Trajectory Generation

From the maps described above, trajectories are generated in several steps. In the first step, the ordered list of waypoints that are used to direct the trajectory are established. This is done by first selecting a starting waypoint, and continuing by using the graph of waypoints and feasible paths. The next waypoint is randomly chosen with a weighting that ensures that new waypoints in the graph are visited before old waypoints are revisited. After the list of waypoints has been generated, the positions at each time interval are created. Using Gauss-Markov processes with a positive mean value, the velocity of the agent is determined.

The agent's position is propagated in a straight line from one waypoint to the next using the velocity given at a certain time. After this base path has been established,

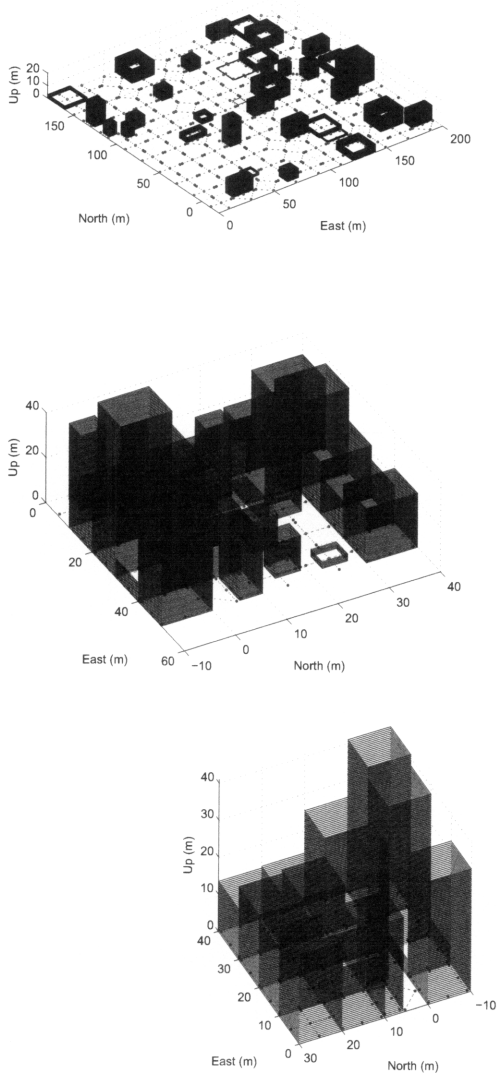

Figure 4-1: Map Scenario Examples for each scenario type: suburban (top), urban (middle), and indoor (bottom).

three Gauss-Markov processes serve to perturb the trajectory from its original path in the three ECEF position components. This is both to ensure the path more realistic as well as make sure that two agents who are in a group, and thus following the same waypoint list with the same velocity, are not directly on top of each other. In the final pass, a boxcar filter serves to smooth the trajectory to ensure that the it only produces small accelerations on the scale of the time step chosen.

After the position component of the trajectories has been generated, the attitudes at each point in time must also be established. The x-axis of the body is first established by pointing towards a position that the agent will be in the next second. The body's y-axis is the cross product of the x-axis of the body and the upward direction in the local frame. The z-axis of the body is such that the body coordinate system is right handed. This makes the z-axis point approximately upwards and the x-axis point approximately towards the direction of motion. Finally, the motion is smoothed to ensure that only small angles are produced on the scale of the time step.

4.2.3 Truth Data Calculation

For the ECEF position p_k, ECEF velocity v_k, body acceleration a_k, attitude directional cosine matrix A_k, and body angular rate ω_k, the truth values must be generated in order to run the optimization. These states are used in generating the state propagation and measurement matrices as described in Section 3.4. These values are calculated from the position and attitudes established by the trajectory generation procedure.

$$(\omega_k \times) = I - A_k A_{k-1}^{-1} \tag{4.1}$$

$$v_k = (p_k - p_{k-1})/dt \tag{4.2}$$

$$a_k = A_{k-1}(v_k - v_{k-1})/dt \tag{4.3}$$

After the body accelerations and angular rates are calculated, the forces due to the difference between the ECEF frame and the inertial frame are added. For the

angular rates, this requires the introduction of an additive vector ω_E due to earth rotation. In the case of the accelerations, two forces, the Coriolis force F_{cor} and the centrifugal forces F_{cent} are also added. The force of gravity F_{grav} is also seen in the body frame, and must also be added. Uncorrupted sensor measurement values for the accelerometers dv and gyroscopes $d\theta$ are then calculated.

$$d\theta_k = (\omega + \omega_E)dt \tag{4.4}$$

$$dv_k = (a_k + F_{cor} + F_{cent} + F_{grav})dt \tag{4.5}$$

In addition to the kinematic state's truth, the error states of the sensors must also be calculated. Given the linear Gaussian models previously introduced, the value of the error states at each time step are stochastically generated. The GPS drift rate and bias, baroaltimeter bias, and IMU time correlated biases are generated in this way.

4.2.4 Pseudorange Measurements and Availability

Pseudorange availability is calculated given a trajectory and the buildings present in a map. The building model for this map is constrained, only allowing for square shaped buildings in line with the ENU coordinate system. Thus, the location of the building's Northwest corner, its two horizontal dimensions, and its height fully define the building model.

In order to check the availability of a pseudorange from a satellite to an agent at a specific time, the ephemeris of the satellite is used to generate the ECEF position of the satellite at the given time. This is then translated into a local spherical coordinate system of the agent, azimuth, elevation, and range. Given the azimuth value of the satellite, the buildings that are in that azimuthal direction from the agent are calculated. The height of the building as seen at the azimuthal direction of the satellite and the range to that building is used to determine the elevation range that can be seen. If the satellite is below the given elevation, the signal is considered to be

occluded by the building. Additionally, satellites below the threshold elevation of 10 degrees are always eliminated due to the large atmospheric effects on the pseudoranges at the lower elevations. After availability of measurements is determined for the optimization, stochastic measurements are generated using the measurement models described in Section 3.4.

4.3 Kalman Filter for Stochastic Measurements

A Kalman filter is used to verify the results of the covariance propagation in the optimization. This verification of the optimization is a necessary step due to the optimization covariance propagation procedure's linearization around truth data, instead of an estimate. In effect, the optimization procedure calculates the lower bound of the covariance that can be obtained by a Kalman filter. The lower bound is approached only under the circumstance that the mean estimate always yields a similar linearization matrix to the truth data linearization.

The Kalman filter works in a similar way to the covariance propagation, except that a mean estimate is kept as well. The mean estimate is calculated using the Unscented Kalman Filter procedure while fusing inter-agent range measurements. Otherwise, the normal EKF procedures are used for measurement fusion and state propagation. This special case is made for the inter-agent range measurement fusion procedure due to linearization problems encountered without it. As agents are relatively close to each other, an error in the mean estimate of position can lead to considerably different Jacobian matrices. The UKF update for the inter-agent range measurements better accounts for these nonlinearities, and keeps the mean estimate error on the order of its expected value. For covariance propagation in both the optimization and the Kalman filter, the normal EKF methods are used.

4.3.1 Error State Filter Propagation

Instead of a direct Kalman filter, an indirect (or error state) Kalman filter is used [31]. In this filter, IMU integration of position, velocity, and attitude, is done outside

of the filter. The error in these kinematic states is calculated by the Kalman filter using external measurements, and added to the full state after a set of measurement updates. Position and velocity are added to corresponding external full states. Attitude is externally stored in a directional cosine matrix that rotates the body frame into ECEF, with only the small angle rotation along the ECEF axes stored in the filter. The full state attitude is propagated by multiplying the directional cosine matrix with a skew matrix containing the components of the three small angle rotations in the filter.

4.3.2 Optimization Verification

Using the Kalman filter, the results for the verification of two scenarios of optimization are shown. In both scenarios, six agents with no grouping travel around an urban and suburban map, with baro-altimeters on every agent. With no groups, all agents have independent trajectories and therefore do not follow each other throughout the map. The requirement on the localization bound for every agent in the cluster is five meters. Shown in Figures 4-2 through 4-7 are statistics about the optimization covariance propagation and the Kalman filter estimated mean and covariance for the urban test.

In the case of the urban test, the solution to the problem is two IMUs on Agents 1 and 4. Shown in Figure 4-2 are the three components of velocity estimation error, and the magnitude of the estimation error for the velocity vector for a representative agent. The error data is marked as "Est. Velocity Error". The data marked "Pred. Variance" comes from the covariance that is predicted from the optimization procedure given the system model and the IMU configuration. The standard deviation (1-sigma) of each component at every point in time is shown. The data marked "Est. Variance" comes from the Kalman filter's covariance estimate as produced while accepting stochastic data. The expected behavior of the error for the components of velocity is to remain within the 1-sigma bound approximately 68.2% of the time. The graph verifies that the error remains close to the estimated 1-sigma bound. It can also be seen that while the predicted standard deviation is always less than the estimated

standard deviation, the two are nearly identical. The Predicted variance is always smaller than the actual estimated variance as the optimization always used the truth estimate to linearize the measurement functions. This yields the theoretical lower bound for the variance, rather than the values obtained by experimental results.

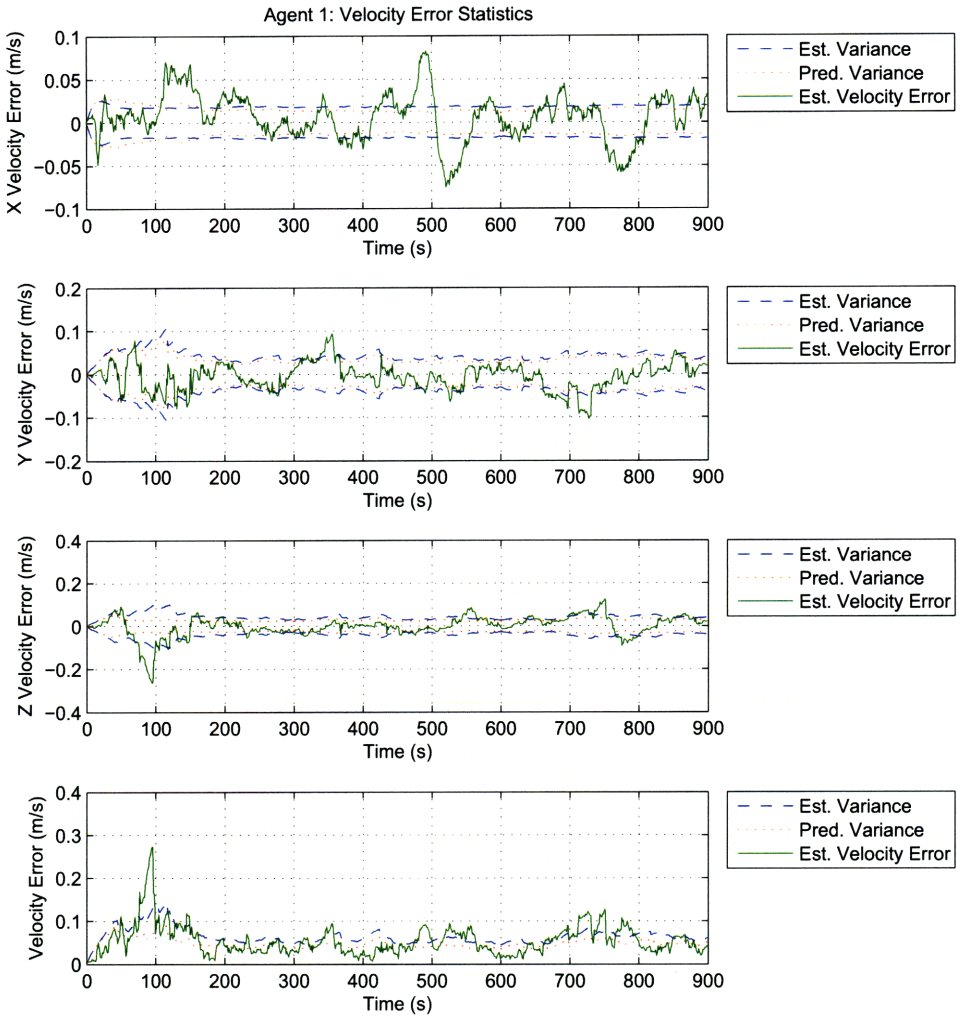

Figure 4-2: Covariance as calculated by optimization procedure and covariance of stochastic simulation for each component and the total velocity. Error from the stochastic simulation is also shown.

Figure 4-3 shows the similar statistics for the position components of the same agent. In the case of the total error, "Pred. Variance" and "Est. Variance" data is computed as the square root of sum of the three components' variances together.

The standard deviation values match as they did in the velocity. However, the error in the components of the position solution for the agent does stray from the standard deviation for a considerable amount of time. The inter-range measurements contribute to these errors in mean and covariance estimation due to their non-linear observation equations. Just as in the velocity, the total error variance is calculated as the square root of the sum of the three components' variances. This random variable is not expected to be Gaussian. In the case where the three components have equal variance, it is a Rayleigh distribution. The single parameter and most likely value of a Rayleigh distribution that is formed as the 2-norm of multiple Gaussian variables is the 2-norm of the Gaussian variables' standard deviations. This Rayleigh distribution parameter is the same as the localization metric that is shown in the plot.

This phenomenon can also be seen in the scaled histogram in Figure 4-4. The histogram shown is that of the error over all time of a single agent, scaled by the estimated variance. This plot is used to show that the error is distributed normally with the estimated variance. The error in an optimal Kalman filter is Gaussian. In this case, the histogram looks like the positive side of a Gaussian PDF with a standard deviation of one and a mean of zero. A slight increase in state values around 2-sigma in the histogram shown is caused by nonlinearities and imperfect state estimation. Figure 4-5 shows the true and estimated trajectory of the representative agent. Tracking of the agent is shown to be occurring correctly. Biases in the position are seen to persist in time, which occurs when measurements are not continuous. However, Figure 4-4 shows that overall, the estimate remains close to the truth trajectory and normally distributed.

Shown in Figure 4-6 are the estimations of the time evolving bias errors on the gyroscopes and accelerometers. In the graphs are the Gauss-Markov processes. The actual estimated mean is shown with standard deviation values above and below it. The true value is shown with respect to the estimated mean and standard deviation. Each of the biases are shown to be within the covariance bound most of the time. Additionally, the predicted and estimated covariances are shown to be equal, except in the case of the Z-axis gyroscope, the variance of which does not decrease as fast,

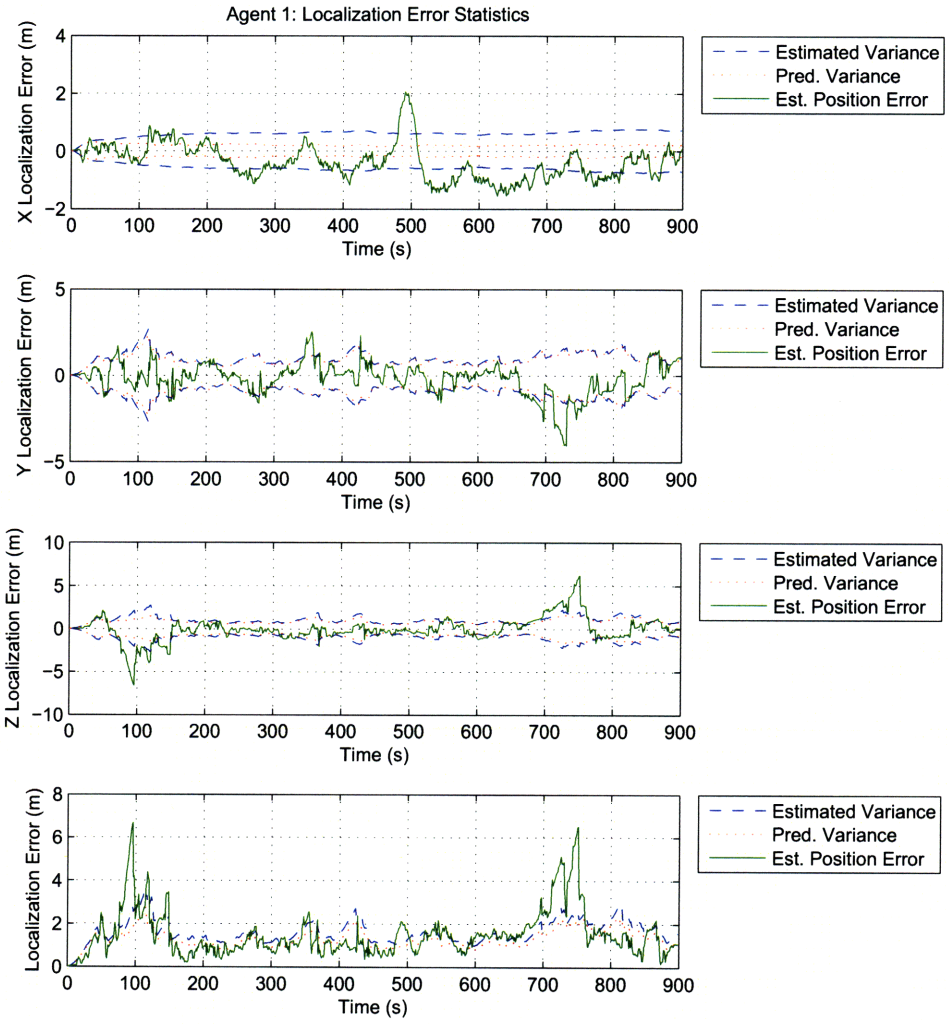

Figure 4-3: Covariance as calculated by optimization procedure and covariance of stochastic simulation for each component and the absolute position. Error from the stochastic simulation is also shown.

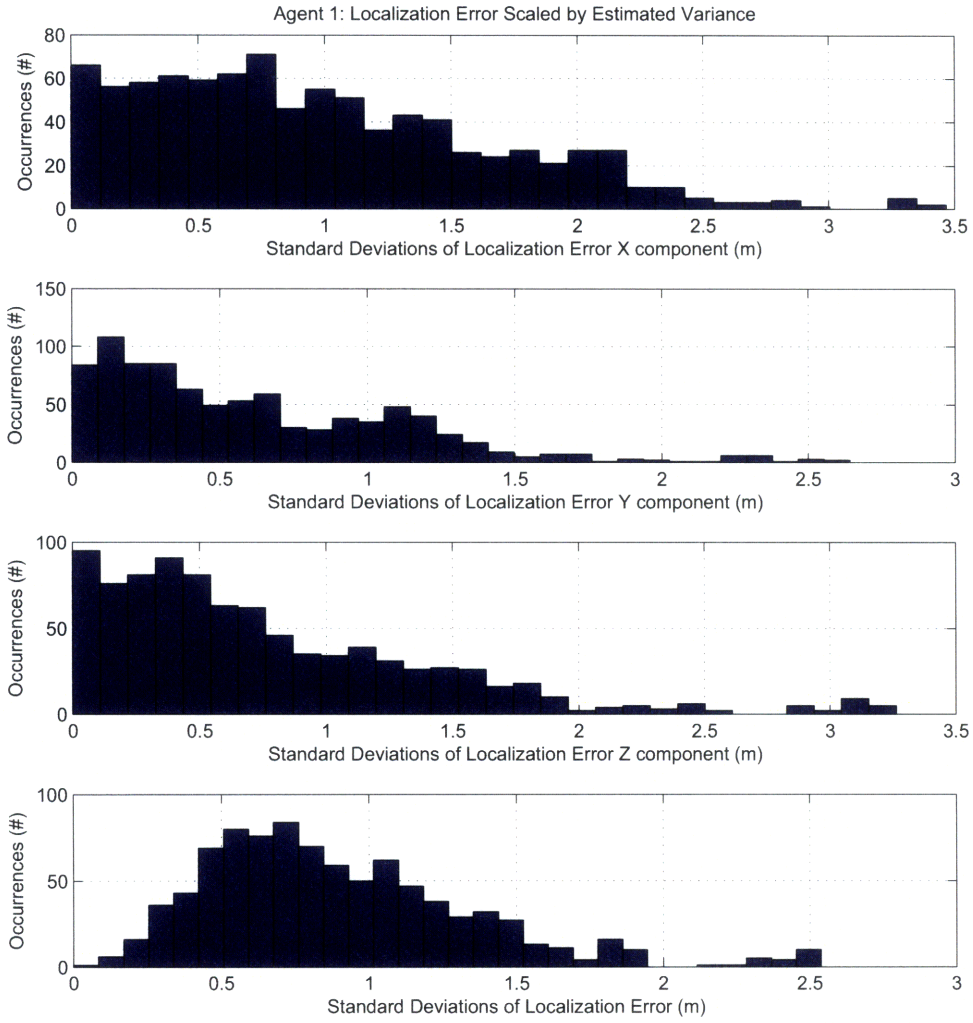

Figure 4-4: Components and total for localization error scaled by the estimated covariance. Error should be normal in each component.

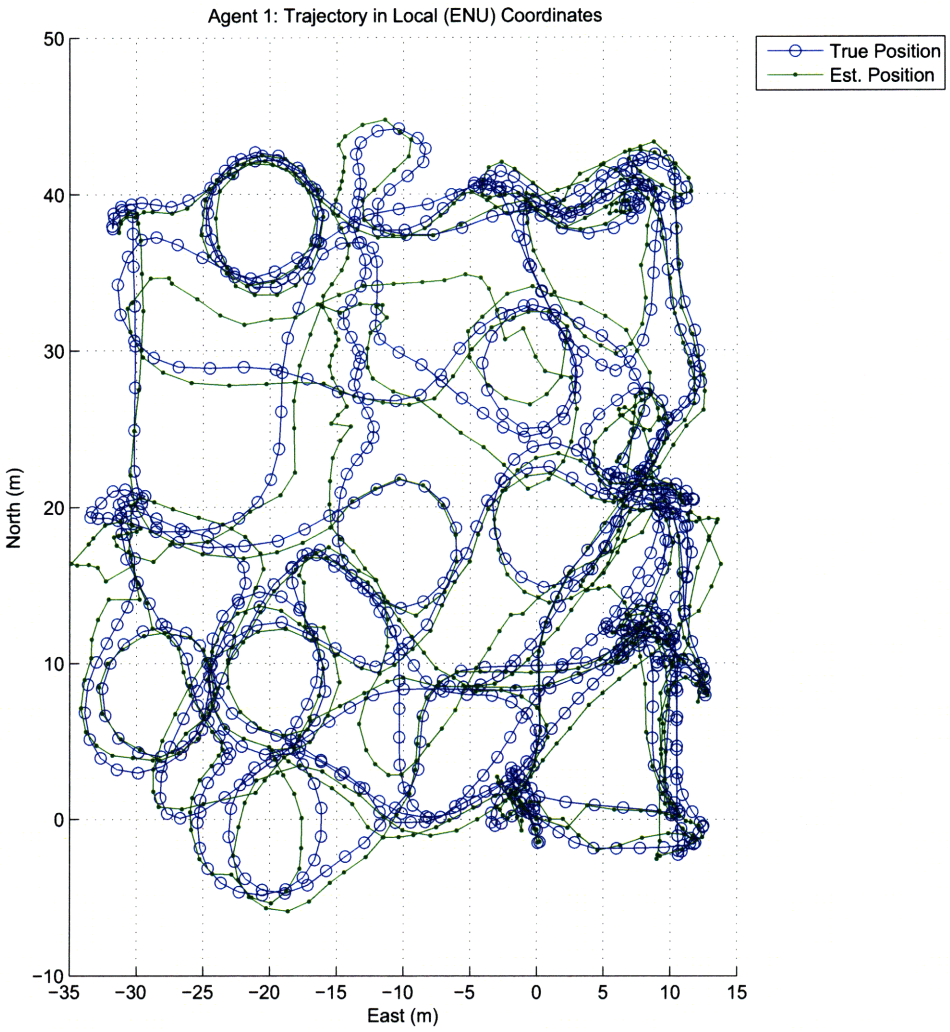

Figure 4-5: The true and estimated localization for a single agent in an urban environment.

but converges to a value matching that predicted by the optimization. This slower convergence may be due to the difference in the estimated and actual attitude. The Z-axis gyroscope is mostly pointing upwards in the local frame. In this orientation, an error in the gyroscope that leads to a drift in heading is not as quickly corrected as is the case for the other two gyroscopes. If either of the other gyroscopes produce an error, it leads to a difference in the estimated and true gravity vector, which in turn produces a large error in position. Errors in the Z-axis gyroscope produce an error in the trajectory that is fixed at a slower rate.

The GPS clock bias, clock drift, as well as the baroaltimeter bias statistics are shown in Figure 4-7. Both the drift and the baroaltimeter bias are Gauss-Markov processes and are shown in a similar way to the IMU errors. The integration of the clock drift and the clock bias estimation error are shown as well. All errors are shown to be as expected. Satellite availability over the entire network throughout the run is shown for reference. "Unique Pranges" refers to the number of satellites from which pseudoranges are received. "Total Pranges" refers to the number of pseudorange measurements at the given time. It can be seen that when the number pseudoranges is low, the localization metric for the representative agent, as well as all of the other agents grow quickly.

Figure 4-8 and Figure 4-9 show the localization metric for every agent over time, as well as the vector magnitude of the error. It can be seen that in the urban scenario, the optimization procedure predicted a value under five meters for the localization metric, but the estimated localization metric produces a value slightly over five meters. However, this discrepancy is relatively small, and all agents are localizing within an expected error around the localization metric. Mean estimates for localization appear to grow quickly when there is little GPS available to the network, but is recovered when more global information is available. In the suburban localization test, the localization metric does not approach the bound of five meters, and the predicted localization versus estimated localization agree. The median value for the localization error is around the localization metric.

Shown in Figure 4-10 and 4-11 are the first minutes of trajectories for represen-

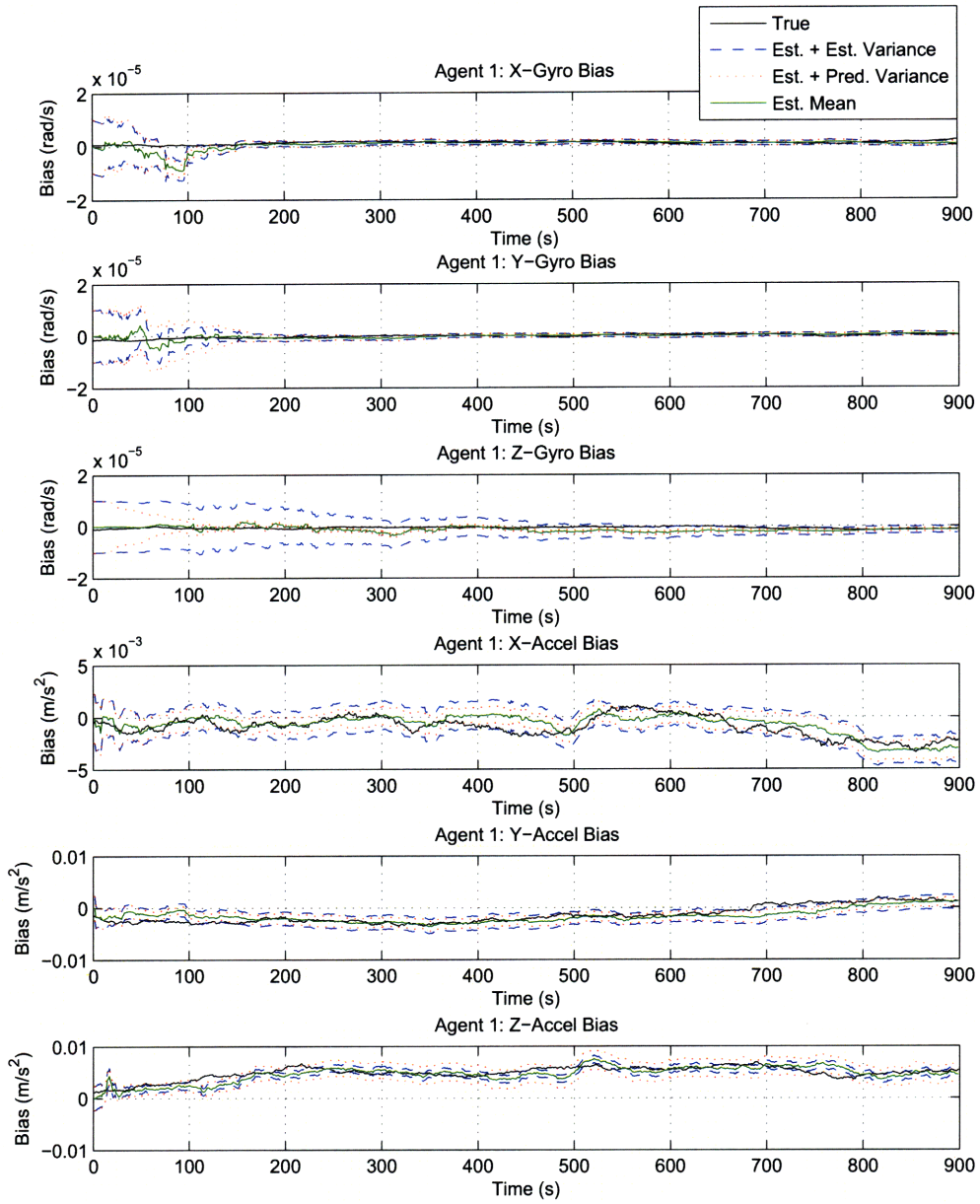

Figure 4-6: Estimates for all error states of the IMU as calculated by optimization procedure and covariance of stochastic simulation. Actual error from stochastic simulation also shown.

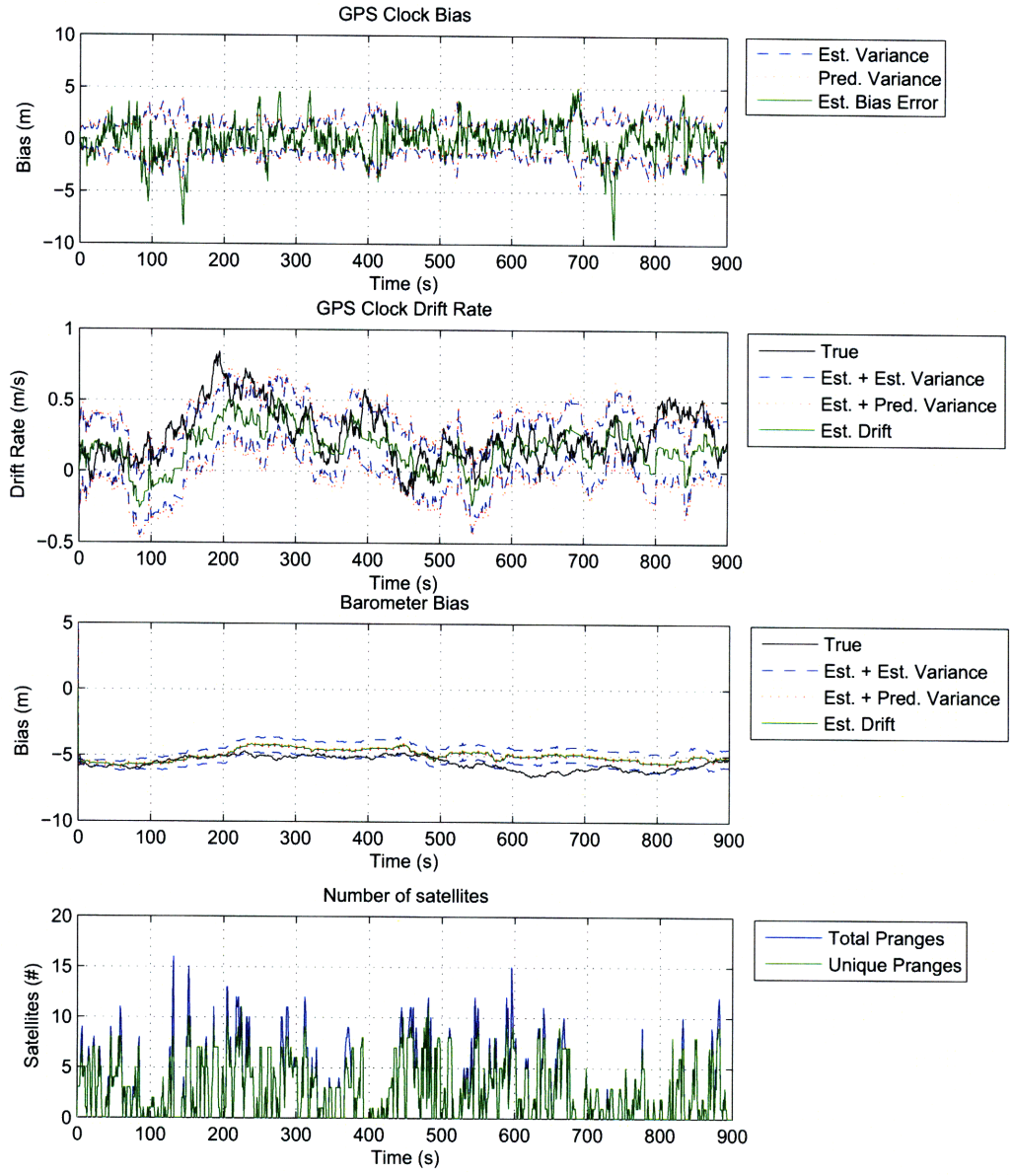

Figure 4-7: Estimates for the the time synchronization error for the entire group of agents as well as the baro-altimeter bias. Number of satellites in view of the entire cluster is also shown.

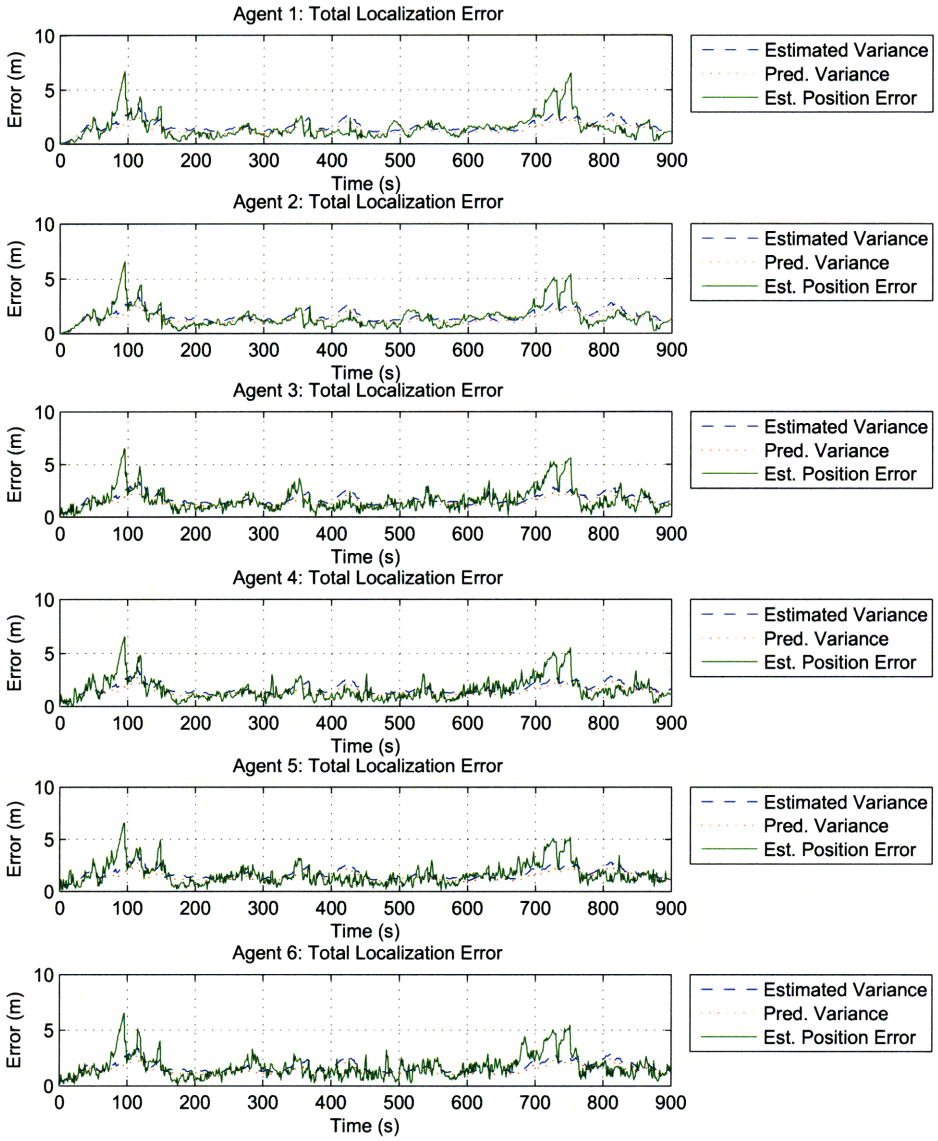

Figure 4-8: Localization metric as calculated by optimization procedure with covariance of stochastic simulation in an urban environment. Error from stochastic simulation is also shown.

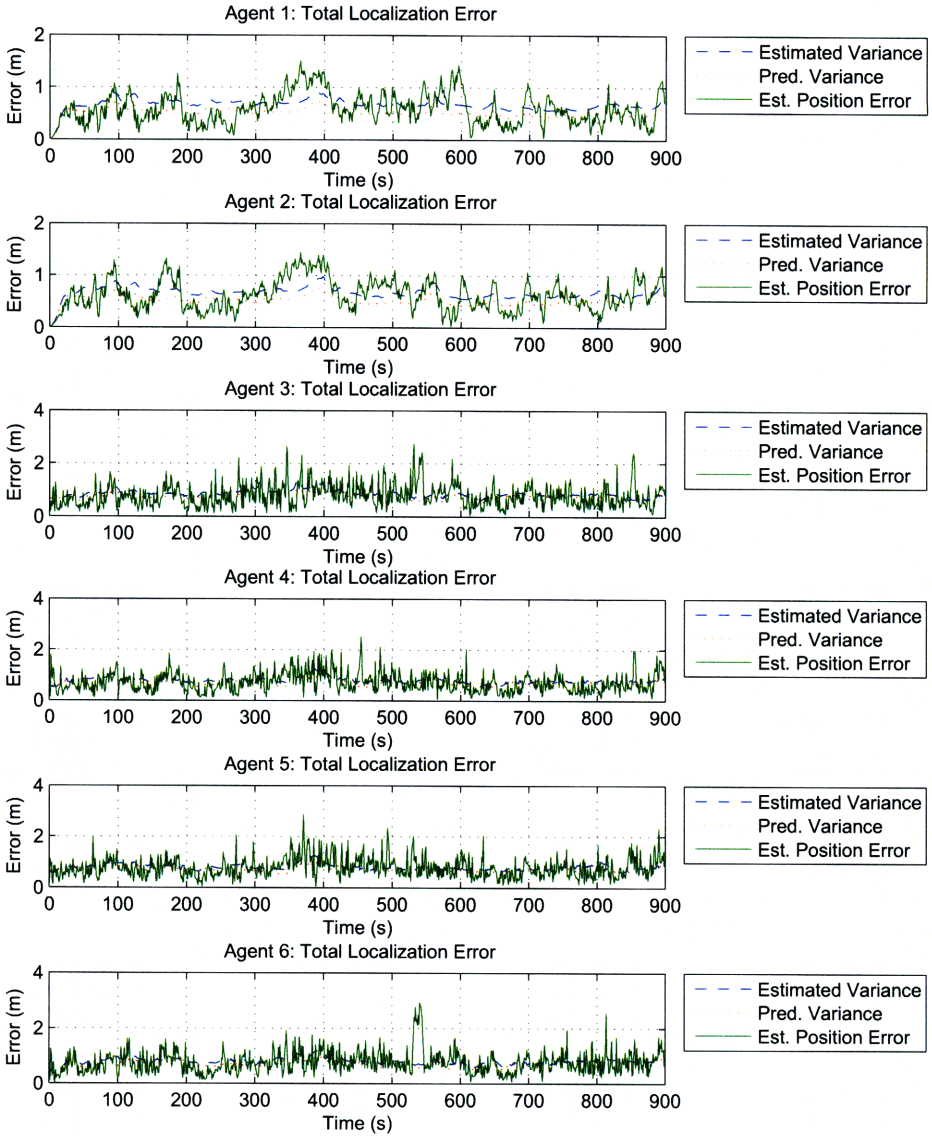

Figure 4-9: Localization metric as calculated by optimization procedure with covariance of stochastic simulation in a suburban environment. Error from stochastic simulation is also shown.

tative agents for urban and suburban localization scenarios, respectively. The lowest cost configuration of IMUs is used. A deviation from the expected error can be seen in the urban trajectory. This can be seen in Figure 4-8 as well, and the error eventually decreases to a nominal level afterwards. Although error growth depends not only on the position of a single agent, it can be seen that when an agent enters a clear area with few surrounding buildings, its estimated error decreases as it gets GPS coverage from more satellites.

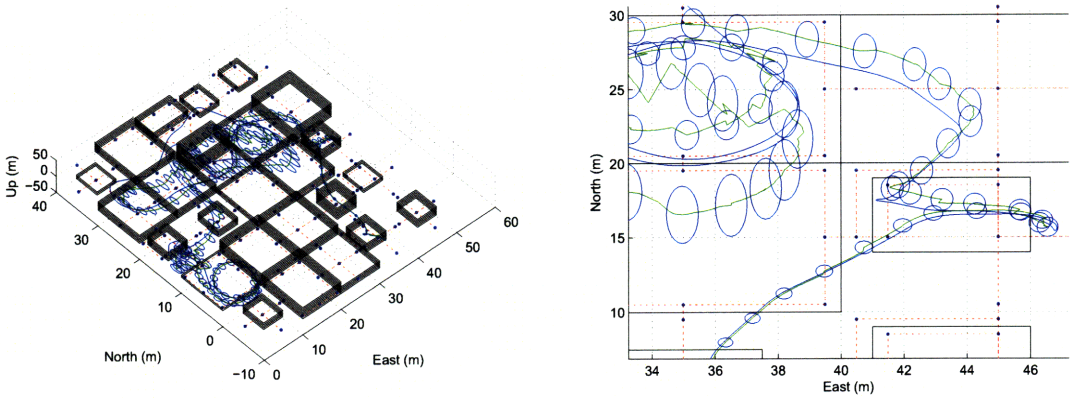

Figure 4-10: Estimated trajectory for one of six agents is shown in green with the urban map and buildings used to generate the trajectories. Ellipses around the estimated trajectory show the 1-sigma errors in the horizontal plane. The true trajectory is shown in blue. Right figure is a magnification of the left figure.

In addition to the lowest cost configurations, the highest cost configurations for the same trajectories are shown in Figure 4-12 and 4-13. The difference between the lowest cost that meets the covariance constraints and the highest cost is minimal, with a significant increase in the cost: four high quality IMUs, or $40,000 in the cost function.

4.4 Scenario Testing

Having defined an optimization problem to solve for the minimally expensive configuration, and verified that the optimization produces a result that a Kalman filter accepting stochastic measurements will also achieve, this section explores the use of

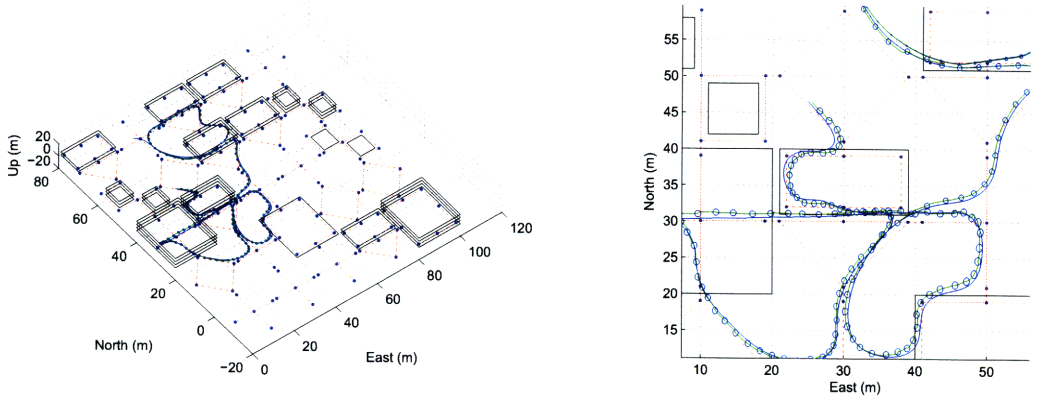

Figure 4-11: Estimated trajectory for one of six agents is shown in green with the suburban map and buildings used to generate the trajectories. Ellipses around the estimated trajectory show the 1-sigma errors in the horizontal plane. The true trajectory is shown in blue. Right figure is a magnification of the left figure.

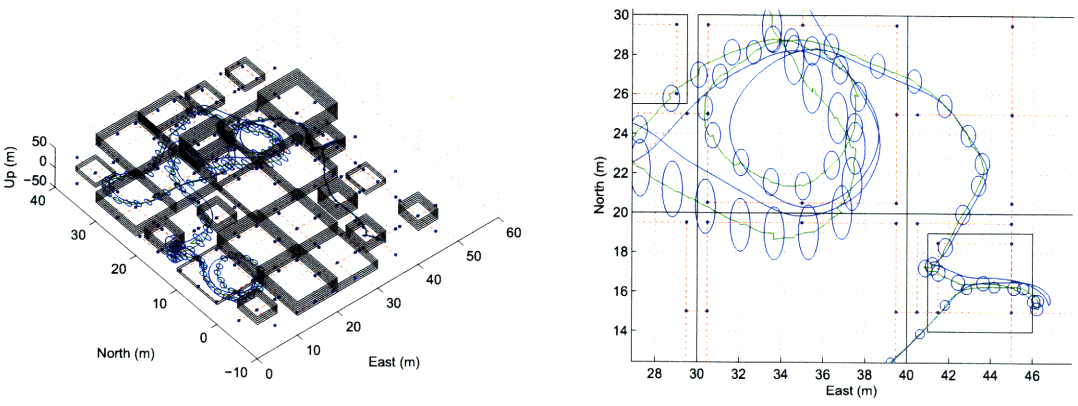

Figure 4-12: Estimated trajectory for one of six agents is shown in green with the urban map and buildings used to generate the trajectories. Ellipses around the estimated trajectory show the 1-sigma errors in the horizontal plane. The true trajectory is shown in blue. Right figure is a magnification of the left figure.

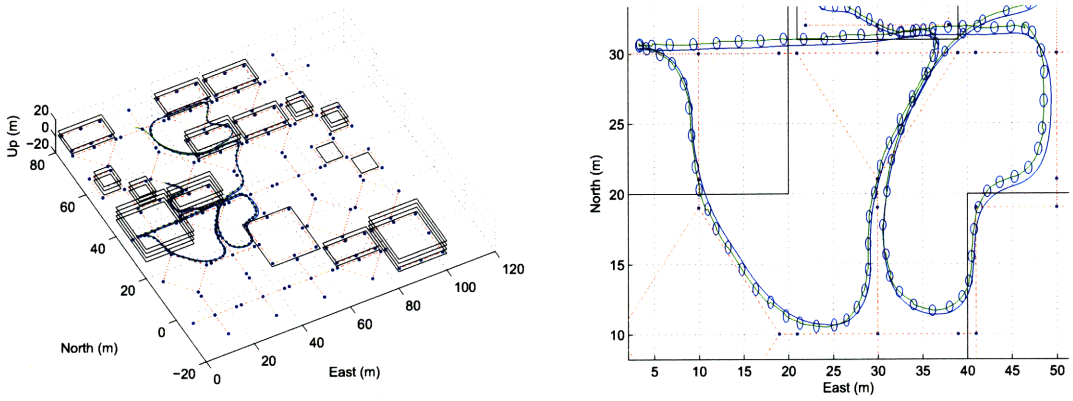

Figure 4-13: Estimated trajectory for one of six agents is shown in green with the suburban map and buildings used to generate the trajectories. Ellipses around the estimated trajectory show the 1-sigma errors in the horizontal plane. The true trajectory is shown in blue. Right figure is a magnification of the left figure.

the optimization procedure to determine the how to instrument a network with sensors in varying environments and conditions. For the IMU assignment problem, the intent is to find the driving factors involved in choosing the number of IMUs that are required for successful tracking of all agents.

4.4.1 Scenario Descriptions

As previously introduced, this thesis examines three different environments for localization: indoor, urban, and suburban. In these environments, six parameters are examined.

1. The effect of baro-altimeters: the effect on IMU placement when baro-altimeters are present on every agent versus when they are not present.

2. Number of groups: while keeping the number of agents the same, agents split into the groups of varying sizes that follow similar trajectories

3. Number of agents: with no groups, the size of the network is changed.

4. Constraining trajectories: some agents are constrained to stay outside while others can roam freely as they normally do.

5. Localization bound: the effect on IMU placement while varying the covariance bound of the localization of every agent.

In all of the listed tests, a base set of parameters for environment and conditions is used. Except when explicitly stated, the network has baro-altimeters available to it, there are no agent groupings, there are six agents in the cluster, agents are not restricted to staying outside, and the localization bound on the optimization algorithm is five meters. Each test consists of changing a small set of those parameters and noting the changes in the required IMU configuration.

In addition to the tests above, the effect on localization while changing the location of the IMUs in the network is examined. Additionally, the benefit of the inter-agent ranging for dead reckoning will be shown by comparing the time that it takes for a single agent with an IMU to reach the bound on covariance versus the time for varying network sizes and IMU availabilities.

4.4.2 Effect of Baro-altimeters

Table 4.2: Optimal Configurations With and Without Baro-Altimeters

Scenario		Agent						Cost
		1	2	3	4	5	6	
Baro	Urban	**HQ**	NO	NO	**HQ**	NO	NO	$20,000
	Suburban	NO	**HQ**	*LQ*	*LQ*	*LQ*	NO	$11,500
No Baro	Urban	NO	**HQ**	**HQ**	**HQ**	NO	NO	$30,000
	Suburban	**HQ**	**HQ**	*LQ*	**HQ**	*LQ*	NO	$31,000

In this test, baro-altimeters are present on all networked agents or none are available at all. In both the urban and suburban environments, the number of high quality IMUs required to localize dropped. In the suburban case, one high quality IMU and three lower quality IMUs are required to localize with a baro-altimeter. A strictly better configuration is required to localize without the altimeter. In the urban case, three IMUs are required for localization without baro-altimeters. The three IMUs preserve the location of three agents in the network through GPS outages. Intuitively, the location of three agents and precise ranging sensors between every agent

can yield the location of all other agents via trilateralization. A baro-altimeter on all agents resolves the z component of every agent, leaving only two degrees of freedom. Two IMUs holding the position error on two agents allow the location of every other agent to be resolved through trilateralization in this scenario.

4.4.3 Number of Agents and Groups

Two factors are examined in the following tests: the total number of agents localizing and sharing information in the network, and the grouping of those agents within the environment. A group of agents travel together and are always on the order of meters away from each other. In both military and emergency response operations, it is most likely the case that agents will operate in groups.

Table 4.3: Optimal Configurations Varying Numbers of Agents (Suburban)

# Agents	Agent											Cost
	1	2	3	4	5	6	7	8	9	10	11	
1	No Solution											N/A
2	**HQ**	**HQ**										$20,000
3	**HQ**	**HQ**	NO									$20,000
4	NO	**HQ**	NO	**HQ**								$20,000
5	NO	**HQ**	NO	NO	**HQ**							$20,000
6	NO	NO	NO	*LQ*	NO	**HQ**						$10,500
7	NO	*LQ*	**HQ**	NO	*LQ*	*LQ*	NO					$11,500
8	NO	NO	*LQ*	NO	**HQ**	NO	NO	NO				$10,500
9	NO	NO	NO	NO	*LQ*	NO	NO	NO	NO			$500
10	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO		$0
11	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	$0

Table 4.3 and 4.4 show the results of varying the number of agents in the network while agents are not grouped together. In both scenarios, a single agent is unable to localize to the precision required by the algorithm. No solution was found by the optimization algorithm, which shows that not even with navigation grade IMUs on every agent was the localization within the required bounds. In the suburban scenario where there are few obstacles to see satellites, two to five agents are able to successfully localize with only two navigation grade IMUs. As the number of agents

increases, the requirements on the number and quality of IMUs decreases until no IMUs are needed at all.

Despite the number of agents increasing, which would intuitively require more information to localize, the GPS information combined by the distributed network decreases the need for aiding IMUs. In the suburban case, not enough global information is available until five agents are sharing pseudoranges and inter-range measurements due to the higher blockage.

Table 4.4: Optimal Configurations Varying Numbers of Agents (Urban)

# Agents	Agent											Cost
	1	2	3	4	5	6	7	8	9	10	11	
1	No Solution											N/A
2	No Solution											N/A
3	No Solution											N/A
4	No Solution											N/A
5	NO	**HQ**	NO	NO	**HQ**							$20,000
6	NO	NO	**HQ**	NO	NO	**HQ**						$20,000
7	**HQ**	NO	NO	**HQ**	NO	NO	NO					$20,000
8	NO	NO	NO	**HQ**	NO	NO	**HQ**	NO				$20,000
9	NO	NO	NO	NO	**HQ**	NO	NO	NO	**HQ**			$20,000
10	NO	NO	NO	NO	NO	**HQ**	NO	NO	NO	**HQ**		$20,000
11	NO	**HQ**	NO	NO	NO	NO	**HQ**	NO	NO	NO	NO	$20,000

As a second test, the number of groups of agents in the network is varied while keeping the total number of agents in the network the same. In table 4.5, the "Agents per Group" column shows the number of distinct base trajectories used by the network. Although each agent in a group deviates from these base trajectories, the amount of the deviation is on the order of one meter. Overall, the effect of grouping agents together is an increase in the number of IMUs required to localize. As the group sizes increase, the cost of instrumenting a network of the same size also increases. Grouping agents together takes away the much of the advantage of increasing the number of agents in the network because every agent in the same group has a similar GPS availability profile. In the urban case, the number of IMUs required, and thus the expense of the network, always goes up with larger group sizes. In the case of the suburban test, this is true except for the case where all agents are in a

single group where the GPS availability for the single group is unusually favorable.

Table 4.5: Optimal Configurations Varying Group Sizes

Scenario	Agents per Group	Agent 1	2	3	4	5	6	Cost
Urban	1	**HQ**	NO	NO	NO	NO	**HQ**	$20,000
	2	**HQ**	**HQ**	NO	NO	NO	**HQ**	$30,000
	3	**HQ**	**HQ**	NO	**HQ**	NO	NO	$30,000
	6	**HQ**	**HQ**	**HQ**	NO	**HQ**	**HQ**	$50,000
Suburban	1	**HQ**	NO	*LQ*	NO	NO	NO	$10,500
	2	NO	**HQ**	NO	NO	**HQ**	NO	$20,000
	3	NO	NO	**HQ**	**HQ**	NO	NO	$20,000
	6	**HQ**	NO	NO	NO	NO	NO	$10,000

4.4.4 Constraining Trajectories

As seen in the previous data sets, it is always harder to localize agents in a network when there are more objects to occlude GPS measurements. Urban scenarios where agents are going inside buildings where all GPS signals are blocked out. Therefore, one consideration may be to constrain a subset of the agents to stay outside of buildings so as to make localization easier. These scenarios should be distinguished from beacon approaches as the agent has the capability to move and is still subject to the GPS blockages of surrounding buildings. Conceptually, an agent that is constrained to remain outside could be an emergency response vehicle, a guard, or a commanding military unit.

Table 4.6 shows the optimal configurations as calculated by the optimization routine for the indoor and urban scenarios. The total number of agents is constant while the number of agents required to stay outside is varied from none to four agents. In the urban case, it can be seen that the requirement of two IMUs does not change, regardless of the number of agents outside. In the urban scenario, it appears that the blockages experienced due to indoor agents is not the sole issue. Even though these agents are outside, they experience blockages from other buildings. In the indoor scenario, agents that are constrained to outside trajectories are able to see all satellites above the elevation mask and not blocked by the buildings. As the outdoor agents

Table 4.6: Optimal Configurations with Agents Constrained to Outside Trajectories

Scenario	Number Outside	Agent 1	2	3	4	5	6	Cost
Urban	0	**HQ**	NO	NO	NO	NO	**HQ**	$20,000
	1	NO	NO	NO	**HQ**	**HQ**	NO	$20,000
	2	NO	NO	**HQ**	NO	NO	**HQ**	$20,000
	3	**HQ**	**HQ**	NO	NO	NO	NO	$20,000
	4	**HQ**	NO	NO	NO	**HQ**	NO	$20,000
Indoor	0	No Solution						N/A
	1	**HQ**	**HQ**	**HQ**	NO	NO	**HQ**	$40,000
	2	**HQ**	**HQ**	NO	NO	NO	**HQ**	$30,000
	3	**HQ**	NO	NO	NO	NO	**HQ**	$20,000
	4	**HQ**	NO	NO	NO	**HQ**	NO	$20,000

were almost never between buildings, they provide many more pseudoranges than in the urban case. Thus, the cost of the configuration drops with the introduction of another agent constrained to be outside.

4.4.5 Sensitivity to Localization Bound

Another tradeoff that is explored is the additional cost for varying location accuracies. In this test, the bound on the localization metric was changed from three meters to 10 meters, keeping the same map and trajectories. As expected, the quality and cost of the IMU configuration decreases with the increase of the localization upper bound. In the suburban test, the localization requires two navigation grade IMUs until five meter localization bound, after which the requirement decreased to no IMU needed at eight meters and higher. In the urban case, the requirement decreased to less than two navigation grade IMUs after eight meters.

4.4.6 Permutations of Sensor Configurations

An important factor to explore in the optimization is whether permutations of IMU configurations change the accuracy of the localization of agents in the network. Two tests are run to determine the effect of permutations on the localization metric of all agents. In the first test, the optimization problem is solved with normal parameters in

Table 4.7: Optimal Configurations Varying Localization Requirements (Suburban)

Localization Bound (m)	Agent						Cost
	1	2	3	4	5	6	
3	**HQ**	NO	NO	**HQ**	NO	NO	$20,000
4	NO	**HQ**	NO	**HQ**	NO	NO	$20,000
5	NO	NO	NO	NO	**HQ**	**HQ**	$20,000
6	**HQ**	*LQ*	*LQ*	*LQ*	NO	NO	$11,500
7	NO	*LQ*	*LQ*	NO	NO	NO	$1,000
8	NO	*LQ*	NO	NO	NO	NO	$500
9	NO	NO	NO	NO	NO	NO	$0
10	NO	NO	NO	NO	NO	NO	$0

Table 4.8: Optimal Configurations Varying Localization Requirements (Urban)

Localization Bound (m)	Agent						Cost
	1	2	3	4	5	6	
3	No Solution						N/A
4	NO	NO	NO	*LQ*	**HQ**	**HQ**	$20,500
5	NO	NO	**HQ**	NO	**HQ**	NO	$20,000
6	**HQ**	NO	NO	NO	NO	**HQ**	$20,000
7	NO	NO	NO	**HQ**	**HQ**	NO	$20,000
8	NO	**HQ**	NO	**HQ**	NO	NO	$20,000
9	**HQ**	*LQ*	*LQ*	NO	NO	NO	$11,000
10	NO	*LQ*	NO	NO	**HQ**	NO	$10,500

an urban environment. The result is a minimum cost configuration of two navigation grade IMUs. All unique permutations of these two IMUs across the network are tested. For each networked agent, the maximum and minimum localization metrics over all configurations are kept and compared to the localization metric of the configuration provided by the optimization algorithm. Figure 4-14 on the left shows the localization metric of the best, worst, and chosen configurations for each agent. The localization metric varies on the order of 10% over all possible permutations of the two navigation grade IMUs, making the assignments almost equivalent.

The optimization problem is also solved for the suburban case, which yields one high quality IMU and one low quality IMU. In the same manner, every permutation of this solution is tested to find the maximum and minimum configuration for every agent. Shown on the right in Figure 4-14 are the results. For each agent, the best localization metric and the worst localization metric over all permutations is shown. The results of this test vary widely, and the configurations cannot be considered equivalent.

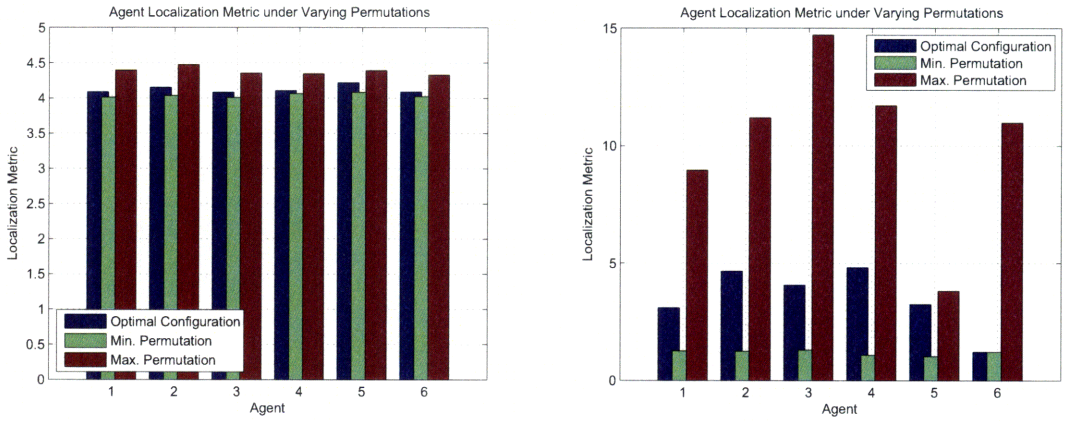

Figure 4-14: Effect of Permutation on Agents' Localization Metrics. Two high quality IMUs' positions are changed to various nodes (left). Two IMUs with different qualities are changed to various nodes (right).

4.4.7 Benefits of Ranging and IMU

In order to test the additional benefit of using UWB ranging measurements as opposed to agents independently locating and tracking themselves, different sizes of groups

with varying qualities of IMUs are tested for the amount of time that they can stay below the five meter localization metric during covariance propagation without any external measurements such as GPS or baro-altimeters. It is shown in Figure 4-15 that when the information provided by several IMUs is linked by UWB ranging between them, the covariance does not grow as fast and the configuration can survive longer under the localization bound. The amount of time that the agents can stay below the localization bound is referred to as the hold time.

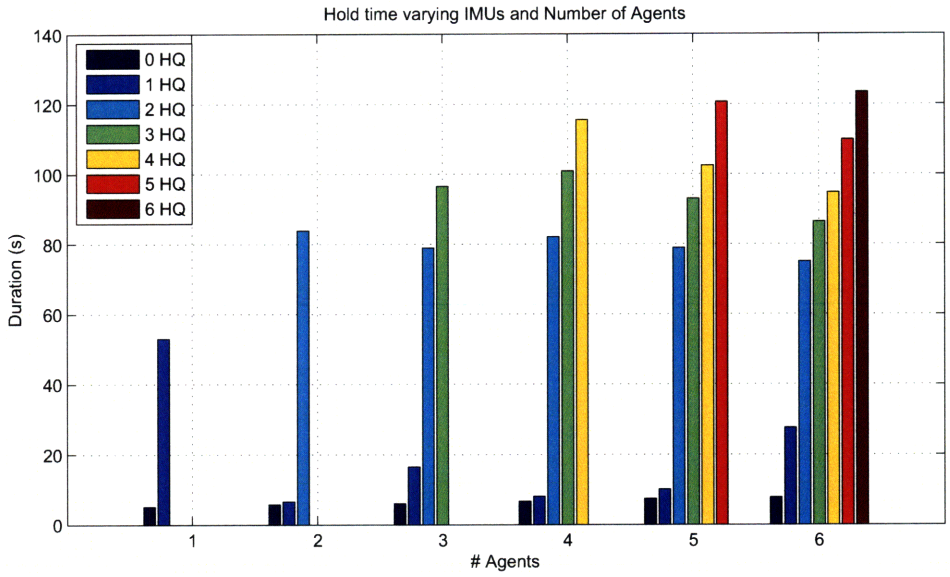

Figure 4-15: GPS outage duration with inter-user ranging, IMUs, and barometers. Varying the number of high quality IMUs in the cluster changes the duration under which localization stays within the bound.

Figure 4-15 shows that a network of six agents instrumented with navigation grade IMUs lasts 2.33 times more than a single agent alone. Given the same number of navigation grade IMUs and more agents in the network, the hold time generally decreases. Additionally, as the number of agents in the network increases, it appears that the benefit of UWB for each additional agent slows. An important attribute to notice is that the hold time greatly increases after adding the second high quality IMU to the network for all numbers of agents above one. This is due to the two IMUs and the baro-altimeters limiting the error growth on the six degrees of freedom of the entire network. The two IMUs additionally fix the three position displacements and

two degrees of rotational freedom of the entire network. The baro-altimeters then constrain the third possible rotation, fixing all six degrees of freedom of the entire network in space. Within the network, the complete inter-agent ranging graph creates a rigid structure that has no internal degrees of freedom.

Chapter 5

Conclusion and Future Work

This chapter, which concludes the thesis, presents the contributions made, as well as a discussion of future work. Section 5.1 discusses possible improvements to be explored on the optimization search strategy as shown in Chapter 3. Section 5.2 details other systems to which the optimization can be used to study. Section 5.3 concludes the thesis with an enumeration of the contributions and work presented.

5.1 Future Work in Optimization

This thesis defines an optimization problem that solves for an minimally expensive instrumentation for a sensor network while maintaining a bound on localization error. A solution strategy is described, and several methods of searching the problem space are introduced. These methods utilize configuration relationships to lessen the time required to find an optimal solution. It is promising that the computation time required for this algorithm can be improved upon with thought on several issues.

1. Accounting for results of covariance propagation: Except for excluding previously eliminated configurations, the process does not account for the outcomes of previous covariance tests. Information such as which covariance bound was not met could be used to determine future test configurations. Additional structure in the problem of the link between elements in the covariance matrix and the specific sensors in the instrumentation could be used.

2. Using faster covariance propagation methods: A method for covariance propagation without matrix inversion is available, but requires that the covariance matrix be factorized into two matrices $P = AB^{-1}$ where A and B are calculated with a linear equation from their previous values [27]. Given that this optimization requires the evaluation of the covariance matrix at every step in time, this method was not initially considered. However, given the knowledge of where previous configurations passed the covariance bound and the superiority relationships between certain configurations, it may not be necessary to test the covariance at every step. Thus, this faster propagation technique could be used.

3. Better sampling strategies for test configurations: the most time efficient method shown in this thesis uses a random sampling procedure to find candidate test instrumentations and how many other instrumentations they would eliminate. Instead of randomly sampling, other strategies for searching the space can be used.

4. Method for counting or iterating through reduced search space: in the sampling case, this thesis used a data structure to keep track of whether an instrumentation had been eliminated by a previous test. If the smaller search space could be iterated through using only the previously tested configurations and their outcomes, the space complexity of the problem is significantly lowered.

5.2 Future Work in Sensor Instrumentation

The verification of the covariance analysis and stochastic simulations for the IMU assignment problem as presented in this work has not been verified with the fusion of real sensor data. Given this data, the sensor models used may require changes. For example, a better model for UWB blockage, biases, and noise in urban areas would be required to obtain results that would be more realistic.

Additionally, the problem of IMU placement within a network of sensors is only

one of many other applicable systems. The solution strategies presented can be used to solve sensor configuration problems where ever there are tiered choices for multiple types of sensors. One area of interest is the localization of a single agent with multiple sensor types available. The optimization algorithm, as well as the relationships between sensor configurations can be used to determine the benefit of a particular sensor's availability during a specific mission.

5.3 Conclusion

This thesis has shown how to solve the problem of instrumenting a group of communicating agents with a variety of sensors available to them for geolocation. Given this network of agents capable of inter-user range measurement and the sharing of additional sensor information, a method is shown to find the minimal cost sensor configuration required to successfully localize every agent in a global coordinate frame.

In Chapter 2, the concepts used to both formulate and solve the problem presented in this thesis are introduced. The Kalman filter is introduced from its basis in probability theory, and two additional Gaussian belief filters, the Information and Unscented filters, are introduced. Using these concepts, an optimization problem is defined in Chapter 3 that minimizes the cost of the sensor configuration for a system while requiring a bound on the covariance. A framework for the solution to the problem is introduced, and various methods for searching through the configuration space are introduced. The optimization is applied to the problem of IMU placement for a group of agents with inter-agent ranging measurements between them.

Chapter 4 explores the IMU placement problem, varying the environments and testing different optimization parameters. Using the optimization procedure, urban, suburban, and indoor environments are tested, along with the addition of baro-altimetry, the effect on the number of agents in the network and the grouping of their trajectories, and the effect of changing the required localization metric for every agent. After these tests, the placement of equal cost IMU configurations and its effect on the localization metric of agents in the network is explored. Finally, the

navigation quality for IMU and UWB ranging is explored by determining how much time various network sizes and IMU configurations can remain within the localization error bounds without GPS.

Bibliography

[1] Sam Y. Bae, Ken J. Hayworth, Karl Y. Yee, Kirill Shcheglov, and Dean V. Wiberg. High-performance mems microgyroscope. volume 4755, pages 316–324. SPIE, 2002.

[2] N. Barbour. Inertial components - past, present, and future. In *AIAA Guidance, Navigation, and Control Conference, Montreal, Canada*, 2001.

[3] C. L. Bennett and G. F. Ross. Time-domain electromagnetics and its applications. *Proceedings of the IEEE*, 66(3):299–318, 1978.

[4] Dimitri P. Bertsekas and John N. Tsitsiklis. *Introduction to Probability.* Athena Scientific, 2002.

[5] T. J. Brand and Phillips. R. E. Foot-to-foot range measurement as an aid to personal navigation. In *Proceedings, ION AM*, pages 113–121, Albuquerque, NM, June 2003.

[6] Brown, R.G. *Introduction to Random Signal Analysis and Kalman Filtering.* John Wiley and Sons, 1983.

[7] W. W. Chow, J. Gea-Banacloche, L. M. Pedrotti, V. E. Sanders, W. Schleich, and M. O. Scully. The ring laser gyro. *Reviews of Modern Physics*, 57:61–104, January 1985.

[8] Federal Communications Commission. First report and order. 02, 48. FCC, 2002.

[9] Joseph Djugash, Sanjiv Singh, George A Kantor, and Wei Zhang. Range-only slam for robots operating cooperatively with sensor networks. In *IEEE International Conference on Robotics and Automation*, pages 2078 – 2084, May 2006.

[10] Arnaud Doucet, Simon Godsill, and Christophe Andrieu. On sequential monte carlo sampling methods for bayesian filtering. *Statistics and Computing*, 10(3):197–208, July 2000.

[11] W. Flenniken, J. Wall, and D. Bevly. Characterization of various imu error sources and the effect on navigation performance. In *ION GNSS Conference*, 2005.

[12] R.J. Fontana. Experimental results from an ultra wideband precision geolocation system. In *Proceedings of the Ultra-wideband, Short-Pulse Electromagnetics 5, Conference*, Gaithersburg, MD, 2000. Kluwer Academeic/Plenum Publishers.

[13] D. Fox, W. Burgard, H. Kruppa, and S. Thrun. A probabilistic approach to collaborative multi-robot localization. *Autonomous Robots*, 2000.

[14] A. Gelb. *Applied optimal estimation.* MIT Press, 1974.

[15] R. Hopkins. The pendulous integrating gyroscope accelerometer (piga) from the v-2 to trident d5, the strategic instrument of choice. In *AIAA Guidance, Navigation, and Control Conference, Montreal, Canada*, 2001.

[16] D.B. Jourdan, Jr. Deyst, J.J., M.Z. Win, and N. Roy. Monte carlo localization in dense multipath environments using uwb ranging. *Ultra-Wideband, 2005. ICU 2005. 2005 IEEE International Conference*, pages 314–319, Sept. 2005.

[17] S. Julier and J. Uhlmann. A new extension of the Kalman filter to nonlinear systems. In *Int. Symp. Aerospace/Defense Sensing, Simul. and Controls, Orlando, FL*, 1997.

[18] R. E. Kalman. A new approach to linear filtering and prediction problems. *Transactions of the ASME Journal of Basic Engineering*, 82 (Series D):35–45, 1960.

[19] E. Kaplan. *Understanding GPS: Principles and Applications.* Artech House, 1996.

[20] J.H. Kim and S. Sukkarieh. A baro-altimeter augmented ins/gps navigation system for uninhabited areial vehicle. In *International Symposium on Satellite Navigation Technology Including Mobile Positioning and Location Services*, Melbourne, Australia, 2003. Australian Global Positioning Systems Society Inc.

[21] A.I. Mourikis and S.I. Roumeliotis. Optimal sensor scheduling for resource-constrained localization of mobile robot formations. *Robotics, IEEE Transactions on [see also Robotics and Automation, IEEE Transactions on]*, 22(5):917–931, Oct. 2006.

[22] L.E. Navarro-Serment, C.J.J. Paredis, and P.K. Khosla. A beacon system for the localization of distributed robotic teams. In *Proceedings of the International Conference on Field and Service Robotics*, Pittsburgh, PA, August 1999.

[23] L. Ojeda and J. Borenstein. Non-GPS navigation with the personal dead-reckoning system. In *Unmanned Systems Technology IX. Edited by Gerhart, Grant R.; Gage, Douglas W.; Shoemaker, Charles M.. Proceedings of the SPIE, Volume 6561, pp. 65610C (2007).*, volume 6561 of *Presented at the Society of Photo-Optical Instrumentation Engineers (SPIE) Conference*, May 2007.

[24] Edwin Olson, John J. Leonard, and Seth Teller. Robust range-only beacon localization. *Oceanic Engineering, IEEE Journal of*, 31(4):949–958, Oct. 2006.

[25] Kaveh Pahlavan, Xinrong Li, Mika Ylianttila, Ranvir Chana, and Matti Latva-aho. An overview of wireless indoor geolocation techniques and systems. In *NETWORKING '00: Proceedings of the IFIP-TC6/European Commission International Workshop on Mobile and Wireless Communication Networks*, pages 1–13, London, UK, 2000. Springer-Verlag.

[26] Judea Pearl. *Probabilistic Reasoning in Intelligent Systems : Networks of Plausible Inference*. Morgan Kaufmann, September 1988.

[27] Samuel J. Prentice. Robust Range-Based Localization and Motion Planning Under Uncertainty Using Ultra-Wideband Radio. Master's thesis, Massachusetts Institute of Technology, Cambridge, MA, September 2007.

[28] J. Rantakokko, P. Hndel, F. Eklf, B. Boberg, M. Junered, D. Akos, I. Skog, H. Bohlin, F. Neregrd, F. Hoffmann, D. Andersson, M. Jansson, and P. Stenumgaard. Positioning of emergency personnel in rescue operations. Technical report, Skolan fr Elektro-och Systemteknik, 2007.

[29] J.K. Ray, M.E. Cannon, and P. Fenton. Gps code and carrier multipath mitigation using a multiantenna system. *Aerospace and Electronic Systems, IEEE Transactions on*, 37(1):183–195, Jan 2001.

[30] K. Reif, S. Gunther, E. Yaz, and R. Unbehauen. Stochastic stability of the discrete-time extended kalman filter. *Automatic Control, IEEE Transactions on*, 44(4):714–728, Apr 1999.

[31] S.I. Roumeliotis, G.S. Sukhatme, and G.A. Bekey. Circumventing dynamic modeling: evaluation of the error-state kalman filter applied to mobile robot localization. In *IEEE International Conference on Robotics and Automation, 1999. Proceedings. 1999*, 1999.

[32] P. G. Savage. Strapdown inertial navigation integration algorithm design part 1: Attitude algorithms. *Journal of Guidance, Control, and Dynamics*, 21(1), 1998.

[33] P. G. Savage. Strapdown inertial navigation integration algorithm design part 2: Velocity and position algorithms. *Journal of Guidance, Control, and Dynamics*, 21(1), 1998.

[34] B. Stoneberger. *Combat Leader's Field Guide*. Stackpole Books, 2005.

[35] Simon Tong and Daphne Koller. Support vector machine active learning with applications to text classification. *J. Mach. Learn. Res.*, 2:45–66, 2002.

[36] T. Tveit. On the complexity of matrix inversion. Mathematical Note, Norwegian University of Science and Technology, November 2003.

[37] L.-P. Wang, K. Deng, L. Zou, R. Wolf, R. J. Davis, and S. Trolier-McKinstry. Microelectromechanical systems (MEMS) accelerometers using lead zirconate titanate thick films. *IEEE Electron Device Letters*, 23:182–184, April 2002.

[38] M.Z. Win, G. Chrisikos, and N.R. Sollenberger. Performance of rake reception in dense multipath channels: implications of spreading bandwidth and selection diversity order. *Selected Areas in Communications, IEEE Journal on*, 18(8):1516–1525, Aug 2000.

[39] D.P. Young, C.M. Keller, D.W. Bliss, and K.W. Forsythe. Ultra-wideband (uwb) transmitter location using time difference of arrival (tdoa) techniques. *Signals, Systems and Computers, 2003. Conference Record of the Thirty-Seventh Asilomar Conference on*, 2:1225–1229 Vol.2, Nov. 2003.